
boto Documentation

Release HEAD

Mitch Garnaat

Sep 17, 2018

Contents

| | | |
|----------|-------------------------------------|-----------|
| 1 | Getting Started | 3 |
| 2 | Currently Supported Services | 5 |
| 3 | Additional Resources | 9 |
| 4 | Release Notes | 11 |
| 4.1 | boto v2.49.0 | 11 |
| 4.2 | boto v2.48.0 | 11 |
| 4.3 | boto v2.47.0 | 12 |
| 4.4 | boto v2.46.1 | 12 |
| 4.5 | boto v2.45.0 | 12 |
| 4.6 | boto v2.44.0 | 12 |
| 4.7 | boto v2.43.0 | 13 |
| 4.8 | boto v2.42.0 | 13 |
| 4.9 | boto v2.41.0 | 13 |
| 4.10 | boto v2.40.0 | 14 |
| 4.11 | boto v2.39.0 | 14 |
| 4.12 | boto v2.38.0 | 15 |
| 4.13 | boto v2.37.0 | 15 |
| 4.14 | boto v2.36.0 | 16 |
| 4.15 | boto v2.35.2 | 16 |
| 4.16 | boto v2.35.1 | 17 |
| 4.17 | boto v2.35.0 | 17 |
| 4.18 | boto v2.34.0 | 18 |
| 4.19 | boto v2.33.0 | 19 |
| 4.20 | boto v2.32.1 | 20 |
| 4.21 | boto v2.32.0 | 21 |
| 4.22 | boto v2.31.1 | 23 |
| 4.23 | boto v2.31.0 | 23 |
| 4.24 | boto v2.30.0 | 24 |
| 4.25 | boto v2.29.1 | 24 |
| 4.26 | boto v2.29.0 | 25 |
| 4.27 | boto v2.28.0 | 25 |
| 4.28 | boto v2.27.0 | 26 |
| 4.29 | boto v2.26.1 | 27 |
| 4.30 | boto v2.26.0 | 27 |

| | | |
|------|---|----|
| 4.31 | boto v2.25.0 | 28 |
| 4.32 | boto v2.24.0 | 29 |
| 4.33 | boto v2.23.0 | 30 |
| 4.34 | boto v2.22.1 | 31 |
| 4.35 | boto v2.22.0 | 32 |
| 4.36 | boto v2.21.2 | 32 |
| 4.37 | boto v2.21.1 | 32 |
| 4.38 | boto v2.21.0 | 33 |
| 4.39 | boto v2.20.1 | 34 |
| 4.40 | boto v2.20.0 | 34 |
| 4.41 | boto v2.19.0 | 35 |
| 4.42 | boto v2.18.0 | 35 |
| 4.43 | boto v2.17.0 | 36 |
| 4.44 | boto v2.16.0 | 36 |
| 4.45 | boto v2.15.0 | 37 |
| 4.46 | boto v2.14.0 | 38 |
| 4.47 | boto v2.13.3 | 39 |
| 4.48 | boto v2.13.2 | 39 |
| 4.49 | boto v2.13.0 | 40 |
| 4.50 | boto v2.12.0 | 41 |
| 4.51 | boto v2.11.0 | 41 |
| 4.52 | boto v2.10.0 | 42 |
| 4.53 | boto v2.9.9 | 43 |
| 4.54 | boto v2.9.8 | 44 |
| 4.55 | boto v2.9.7 | 45 |
| 4.56 | boto v2.9.6 | 46 |
| 4.57 | boto v2.9.5 | 47 |
| 4.58 | boto v2.9.4 | 47 |
| 4.59 | boto v2.9.3 | 48 |
| 4.60 | boto v2.9.2 | 49 |
| 4.61 | boto v2.9.1 | 49 |
| 4.62 | boto v2.9.0 | 50 |
| 4.63 | boto v2.8.0 | 51 |
| 4.64 | boto v2.7.0 | 52 |
| 4.65 | boto v2.6.0 | 54 |
| 4.66 | boto v2.5.2 | 56 |
| 4.67 | boto v2.5.1 | 56 |
| 4.68 | boto v2.5.0 | 57 |
| 4.69 | boto v2.4.0 | 57 |
| 4.70 | boto v2.3.0 | 59 |
| 4.71 | boto v2.2.2 | 60 |
| 4.72 | boto v2.2.1 | 60 |
| 4.73 | boto v2.2.0 | 60 |
| 4.74 | boto v2.1.1 | 62 |
| 4.75 | boto v2.1.0 | 63 |
| 4.76 | Release Notes for boto 2.0 | 65 |
| 4.77 | Major changes for release 2.0b1 | 69 |
| 4.78 | Getting Started with Boto | 69 |
| 4.79 | An Introduction to boto's EC2 interface | 72 |
| 4.80 | Using Elastic Block Storage (EBS) | 75 |
| 4.81 | EC2 Security Groups | 76 |
| 4.82 | An Introduction to boto's Elastic Mapreduce interface | 78 |
| 4.83 | An Introduction to boto's Autoscale interface | 80 |
| 4.84 | CloudFront | 83 |

| | | |
|-------|--|-----|
| 4.85 | An Introduction to boto's SimpleDB interface | 87 |
| 4.86 | An Introduction to boto's DynamoDB interface | 90 |
| 4.87 | An Introduction to boto's RDS interface | 95 |
| 4.88 | An Introduction to boto's SQS interface | 97 |
| 4.89 | Simple Email Service Tutorial | 102 |
| 4.90 | Amazon Simple Workflow Tutorial | 104 |
| 4.91 | An Introduction to boto's Cloudsearch interface | 116 |
| 4.92 | CloudWatch | 122 |
| 4.93 | An Introduction to boto's VPC interface | 124 |
| 4.94 | An Introduction to boto's Elastic Load Balancing interface | 127 |
| 4.95 | An Introduction to boto's S3 interface | 131 |
| 4.96 | An Introduction to boto's Route53 interface | 139 |
| 4.97 | Boto Config | 141 |
| 4.98 | About the Documentation | 149 |
| 4.99 | Contributing to Boto | 150 |
| 4.100 | Command Line Tools | 153 |
| 4.101 | An Introduction to boto's Support interface | 154 |
| 4.102 | An Introduction to boto's DynamoDB v2 interface | 156 |
| 4.103 | Migrating from DynamoDB v1 to DynamoDB v2 | 167 |
| 4.104 | Migrating from RDS v1 to RDS v2 | 173 |
| 4.105 | Applications Built On Boto | 175 |
| 4.106 | Auto Scaling Reference | 175 |
| 4.107 | AWS Lambda | 188 |
| 4.108 | Elastic Beanstalk | 192 |
| 4.109 | boto | 204 |
| 4.110 | cloudformation | 225 |
| 4.111 | CloudFront | 234 |
| 4.112 | CloudHSM | 243 |
| 4.113 | Cloudsearch | 246 |
| 4.114 | Cloudsearch | 262 |
| 4.115 | CloudSearch Domain | 278 |
| 4.116 | CloudTrail | 283 |
| 4.117 | CloudWatch Reference | 288 |
| 4.118 | CodeDeploy | 295 |
| 4.119 | Cognito Identity | 306 |
| 4.120 | Cognito Sync | 312 |
| 4.121 | Config | 316 |
| 4.122 | contrib | 321 |
| 4.123 | Data Pipeline | 321 |
| 4.124 | DynamoDB | 327 |
| 4.125 | DynamoDB2 | 349 |
| 4.126 | EC2 | 403 |
| 4.127 | EC2 Container Service | 464 |
| 4.128 | ECS | 470 |
| 4.129 | Amazon ElastiCache | 471 |
| 4.130 | Elastic Transcoder | 485 |
| 4.131 | ELB Reference | 495 |
| 4.132 | EMR | 506 |
| 4.133 | file | 515 |
| 4.134 | fps | 518 |
| 4.135 | Glacier | 521 |
| 4.136 | GS | 540 |
| 4.137 | IAM | 573 |
| 4.138 | API Reference | 588 |

| | |
|------------------------------------|-----|
| 4.139 KMS | 821 |
| 4.140 CloudWatch Logs | 828 |
| 4.141 Machine Learning | 833 |
| 4.142 Opsworks | 846 |
| 4.143 RDS2 | 874 |
| 4.144 Route 53 Domains | 910 |
| 4.145 SDB DB Reference | 918 |
| 4.146 Support | 926 |
| 4.147 boto v2.46.0 | 932 |
| 4.148 boto v2.xx.x | 932 |
| 4.149 boto v2.xx.x | 933 |
| 4.150 Indices and tables | 933 |

Python Module Index **935**

Note: [Boto3](#), the next version of Boto, is now stable and recommended for general use. It can be used side-by-side with Boto in the same project, so it is easy to start using Boto3 in your existing projects as well as new projects. Going forward, API updates and all new feature work will be focused on Boto3.

An integrated interface to current and future infrastructural services offered by [Amazon Web Services](#).

Currently, all features work with Python 2.6 and 2.7. Work is under way to support Python 3.3+ in the same codebase. Modules are being ported one at a time with the help of the open source community, so please check below for compatibility with Python 3.3+.

To port a module to Python 3.3+, please view our [Contributing Guidelines](#) and the Porting Guide. If you would like, you can open an issue to let others know about your work in progress. Tests **must** pass on Python 2.6, 2.7, 3.3, and 3.4 for pull requests to be accepted.

CHAPTER 1

Getting Started

If you've never used `boto` before, you should read the *Getting Started with Boto* guide to get familiar with `boto` & its usage.

Currently Supported Services

- **Compute**

- *Elastic Compute Cloud (EC2)* – (*API Reference*) (Python 3)
- *Elastic MapReduce (EMR)* – (*API Reference*) (Python 3)
- *Auto Scaling* – (*API Reference*) (Python 3)
- *Kinesis* – (*API Reference*) (Python 3)
- *Lambda* – (*API Reference*) (Python 3)
- *EC2 Container Service (ECS)* – (*API Reference*) (Python 3)

- **Content Delivery**

- *CloudFront* – (*API Reference*) (Python 3)

- **Database**

- *DynamoDB2* – (*API Reference*) – (*Migration Guide from v1*)
- *DynamoDB* – (*API Reference*) (Python 3)
- *Relational Data Services 2 (RDS)* – (*API Reference*) – (*Migration Guide from v1*)
- *Relational Data Services (RDS)* – (*API Reference*)
- *ElastiCache* – (*API Reference*) (Python 3)
- *Redshift* – (*API Reference*) (Python 3)
- *SimpleDB* – (*API Reference*) (Python 3)

- **Deployment and Management**

- *CloudFormation* – (*API Reference*) (Python 3)
- *Elastic Beanstalk* – (*API Reference*) (Python 3)
- *Data Pipeline* – (*API Reference*) (Python 3)
- *Opsworks* – (*API Reference*) (Python 3)

- CloudTrail – (*API Reference*) (Python 3)
- CodeDeploy – (*API Reference*) (Python 3)
- **Administration & Security**
 - Identity and Access Management (IAM) – (*API Reference*) (Python 3)
 - Security Token Service (STS) – (*API Reference*) (Python 3)
 - Key Management Service (KMS) – (*API Reference*) (Python 3)
 - Config – (*API Reference*) (Python 3)
 - CloudHSM – (*API Reference*) (Python 3)
- **Application Services**
 - Cloudsearch 2 – (*API Reference*) (Python 3)
 - *Cloudsearch* – (*API Reference*) (Python 3)
 - CloudSearch Domain – (*API Reference*) (Python 3)
 - Elastic Transcoder – (*API Reference*) (Python 3)
 - *Simple Workflow Service (SWF)* – (*API Reference*) (Python 3)
 - *Simple Queue Service (SQS)* – (*API Reference*) (Python 3)
 - Simple Notification Service (SNS) – (*API Reference*) (Python 3)
 - *Simple Email Service (SES)* – (*API Reference*) (Python 3)
 - Amazon Cognito Identity – (*API Reference*) (Python 3)
 - Amazon Cognito Sync – (*API Reference*) (Python 3)
 - Amazon Machine Learning – (*API Reference*) (Python 3)
- **Monitoring**
 - *CloudWatch* – (*API Reference*) (Python 3)
 - CloudWatch Logs – (*API Reference*) (Python 3)
- **Networking**
 - *Route 53* – (*API Reference*) (Python 3)
 - Route 53 Domains – (*API Reference*) (Python 3)
 - *Virtual Private Cloud (VPC)* – (*API Reference*) (Python 3)
 - *Elastic Load Balancing (ELB)* – (*API Reference*) (Python 3)
 - AWS Direct Connect (Python 3)
- **Payments & Billing**
 - Flexible Payments Service (FPS) – (*API Reference*)
- **Storage**
 - *Simple Storage Service (S3)* – (*API Reference*) (Python 3)
 - Amazon Glacier – (*API Reference*) (Python 3)
 - Google Cloud Storage – (*API Reference*)
- **Workforce**

- Mechanical Turk – (*API Reference*)

- **Other**

- Marketplace Web Services – (*API Reference*) (Python 3)

- *Support* – (*API Reference*) (Python 3)

Additional Resources

- *Applications Built On Boto*
- *Command Line Utilities*
- *Boto Config Tutorial*
- *Contributing to Boto*
- Evaluating Application performance with Boto logging
- Boto Source Repository
- Boto Issue Tracker
- Boto Twitter
- Follow Mitch on Twitter
- Join our IRC channel (#boto on FreeNode).

4.1 boto v2.49.0

date 2018/07/11

Updated the CA Bundle as well as some bucket encryption fixes.

4.1.1 Changes

- Import the latest CA Bundle from certifi ([issue 3818](#), [commit e4699cba](#))
- Fix to support uploads to KMS-encrypted buckets. ([issue 3800](#), [commit 0a1d9040](#))
- Support fetching GCS bucket encryption metadata. ([issue 3799](#), [commit 132b64d2](#))
- Update layer1.py ([issue 3765](#), [commit 53340159](#))
- Fix tests/unit/glacier/test_writer.py to make work with pypy. ([issue 3762](#), [commit 8402c5d6](#))

Bumped to 2.48.0

4.2 boto v2.48.0

date 2017/06/29

This release addresses a few S3 related bugs as well as a bug with the recent endpoint heuristics feature.

4.2.1 Changes

- Fix generate_url() AttributeError when using anonymous connections ([issue 3734](#), [commit 83481807](#))
- Use RegionInfo by default with heuristics ([issue 3737](#), [commit 0a9b1140](#))

- Allow specifying s3 host from boto config file. (issue 3738, commit dcf7512)

Bumped to 2.47.0

4.3 boto v2.47.0

date 2017/05/24

Adds features for Google Cloud Storage.

4.3.1 Changes

- Loosen requirements for ID field in PROJECT_PRIVATE_RE. (issue 3729, commit 5e85d7c7)
- Populate storage class from HEAD Object responses. (issue 3691, commit 315b76e0)

Bumped to 2.46.1

4.4 boto v2.46.1

date 2017/02/20

Fixes a bug where a recently added module was not added to setup.py

4.4.1 Changes

- Add boto.vendored.regions to setup.py (issue 3682, commit 43e796d1)

4.5 boto v2.45.0

date 2016/12/14

Add support for eu-west-2 region.

4.5.1 Changes

- Add support for eu-west-2 (issue 3654, commit 40c68db)

4.6 boto v2.44.0

date 2016/12/08

Adds support for ca-central-1 region and gs object-level storage class.

4.6.1 Changes

- Update endpoints (issue 3649, commit a1eae11)
- Add gs support for object-level storage class features. (issue 3635, commit dc4bf34)

4.7 boto v2.43.0

date 2016/10/17

Adds support for `us-east-2` endpoint.

4.7.1 Changes

- Add support for `us-east-2` endpoint (commit 262ed00)
- Account for metadata update propagation delay (issue 3615, commit 592dae3)
- `boto.dynamodb2.table.Table#batch_get()` fails to paginate results if provisioned throughput is exceeded (issue 3574, commit abb3847)

4.8 boto v2.42.0

date 2016/07/19

Updates the Mechanical Turk API and fixes some bugs.

4.8.1 Changes

- Respect `is_secure` parameter in `generate_url_sigv4` (commit 59ba28d)
- Update MTurk API (issue 3563, commit 250d891)

4.9 boto v2.41.0

date 2016/06/27

Update documentation and endpoints file.

4.9.1 Changes

- Update `endpoints.json` (issue 3564, commit 5e786b4)
- Remove the broken link to PDF's (issue 3562, commit 46ffb0c)

4.10 boto v2.40.0

date 2016/04/28

Fixes several bugs.

4.10.1 Changes

- ryansydnor-s3: Allow s3 bucket lifecycle policies with multiple transitions ([commit c6d5af3](#))
- Fixes upload parts for glacier ([issue 3524](#), [commit d1973a4](#))
- pslawski-unicode-parse-qs: Move utility functions over to compat Add S3 integ test for non-ascii keys with sigv4 Fix quoting of tilde in S3 canonical_uri for sigv4 Parse unicode query string properly in Python 2 ([issue 2844](#), [commit 5092c6d](#))
- ninchat-config-fix: Add `__setstate__` to fix pickling test fail Add unit tests for config parsing Don't access parser through `__dict__` Config: Catch specific exceptions when wrapping ConfigParser methods Config: Don't inherit from ConfigParser ([issue 3474](#), [commit c21aa54](#))

4.11 boto v2.39.0

date 2016/01/18

Add support for ap-northeast-2, update documentation, and fix several bugs.

4.11.1 Changes

- Autodetect sigv4 for ap-northeast-2 ([issue 3461](#), [commit c2a17ce](#))
- Added support for ap-northeast-2 ([issue 3454](#), [commit c3c1ddd](#))
- Remove VeriSign Class 3 CA from trusted certs ([issue 3450](#), [commit 8a025df](#))
- Add note about boto3 on all pages of boto docs ([commit 9bd904c](#))
- Fix for listing EMR steps based on cluster_states filter ([issue 3399](#), [commit 0f92f35](#))
- Fixed param name in set_contents_from_string docstring ([issue 3420](#), [commit e30297b](#))
- Closes #3441 Remove py3 test whitelist Update rds to pass on py3 Update mturk to pass tests on py3 Update cloudsearchdomain tests to work with py3 ([issue 3441](#), [commit 5b2f552](#))
- Run tests against py35 ([commit 7d039d0](#))
- Fix Glacier test failure in python 3.5 due to MagicMock ([issue 3412](#), [commit d042f07](#))
- Undo log message change BF(PY3): use `except ...` as syntax instead of `except ...`, ([commit 607cad7](#))
- Fix travis CI builds for PY3 ([issue 3439](#), [commit 22ab610](#))
- Spelling fixes ([issue 3425](#), [commit f43bbbd](#))
- Fixed docs ([issue 3401](#), [commit 4f66311](#))
- Add deprecation notice to emr methods ([issue 3422](#), [commit cee6159](#))
- Add some GovCloud endpoints ([issue 3421](#), [commit 5afc068](#))

4.12 boto v2.38.0

date 2015/04/09

This release adds support for Amazon Machine Learning and fixes a couple of issues.

4.12.1 Changes

- Add support for Amazon Machine Learning ([commit ab32d572](#))
- Fix issue with modify reserved instances for modifying instance type ([issue 3085](#), [commit b8ea7a04](#))

4.13 boto v2.37.0

date 2015/04/02

This release updates AWS CloudTrail to the latest API to support the `LookupEvents` operation, adds new regional service endpoints and fixes bugs in several services.

Note: The CloudTrail `create_trail` operation no longer supports the deprecated `trail` parameter, which has been marked for removal by the service since early 2014. Instead, you pass each trail parameter as a keyword argument now. Please see the [reference](#) to help port over existing code.

4.13.1 Changes

- Update AWS CloudTrail to the latest API. ([issue 3074](#), [commit bccc29a](#))
- Add support for `UsePreviousValue` to CloudFormation `UpdateStack`. ([issue 3029](#), [commit 8a8a22a](#))
- Fix `BOTH_PATH` to work with Windows drives ([issue 2823](#), [commit 7ba973e](#))
- Fix division calculation in S3 docs. ([issue 3018](#), [commit 4ffd9ba](#))
- Add Boto 3 link in README. ([issue 3013](#), [commit 561716c](#))
- Add more regions for `configservice` ([issue 3009](#), [commit a82244f](#))
- Add `eu-central-1` endpoints (Frankfurt region) for IAM and Route53 ([commit 5ff4add](#))
- Fix unit tests from hanging ([commit da9f9b7](#))
- Fixed wording in dynamodb tutorial ([issue 2993](#), [commit 36cadf4](#))
- Update SWF objects to keep a consistent region name. ([issue 2985](#), [issue 2980](#), [issue 2606](#), [commit ce75a19](#))
- Print archive ID in glacier upload script. ([issue 2951](#), [commit 047c7d3](#))
- Add some minor documentation for Route53 tutorial. ([issue 2952](#), [commit b855fb3](#))
- Add Amazon DynamoDB online indexing support on High level API ([issue 2925](#), [commit 0621c53](#))
- Ensure `Content-Length` header is a string. ([issue 2932](#), [commit 34a0f63](#))
- Correct docs around overriding SGs on ELBs ([issue 2937](#), [commit 84d0ff9](#))
- Fix DynamoDB tests. ([commit 616ee80](#))
- Fix region bug. ([issue 2927](#), [commit b1cb61e](#))

- Fix import for `boto.cloudhsm.layer1.CloudHSMConnection`. (issue 2926, commit 1944d35)

4.14 boto v2.36.0

date 2015/01/27

This release adds support for AWS Key Management Service (KMS), AWS Lambda, AWS CodeDeploy, AWS Config, AWS CloudHSM, Amazon EC2 Container Service (ECS), Amazon DynamoDB online indexing, and fixes a few issues.

4.14.1 Changes

- Add Amazon DynamoDB online indexing support.
- Allow for binary to be passed to sqs message (issue 2913, commit 8af9b42)
- Kinesis update (issue 2891, commit 4874e19)
- Fixed spelling of `boto.awslambda` package. (issue 2914, commit de769ac)
- Add support for Amazon EC2 Container Service (issue 2908, commit 4480fb4)
- Add support for CloudHSM (issue 2905, commit 6055a35)
- Add support for AWS Config (issue 2904, commit 51e9221)
- Add support for AWS CodeDeploy (issue 2899, commit d935356)
- Add support for AWS Lambda (issue 2896, commit 6748016)
- Update both Cognito's to the latest APIs (issue 2909, commit 18c1251)
- Add sts for eu-central-1. (issue 2906, commit 54714ff)
- Update opsworks to latest API (issue 2892, commit aed3302)
- Add AWS Key Management Support (issue 2894, commit ef7d2cd)

4.15 boto v2.35.2

date 2015/01/19

This release adds ClassicLink support for Auto Scaling and fixes a few issues.

4.15.1 Changes

- Add support for new data types in DynamoDB. (issue 2667, commit 68ad513)
- Expose cloudformation `UsePreviousTemplate` parameter. (issue 2843, issue 2628, commit 873e89c)
- Fix documentation around using custom connections for DynamoDB tables. (issue 2842, issue 1585, commit 71d677f)
- Fixed bug that unable call `query_2` after call `describe` method on `dynamodb2` module. (issue 2829, commit 66addce)

4.16 boto v2.35.1

date 2015/01/09

This release fixes a regression which results in an infinite while loop of requests if you query an empty Amazon DynamoDB table.

4.16.1 Changes

- Check for results left after computing `self._keys_left` (issue 2871, commit d3c2595)

4.17 boto v2.35.0

date 2015/01/08

This release adds support for Amazon EC2 Classic Link which allows users to link classic instances to Classic Link enabled VPCs, adds support for Amazon CloudSearch Domain, adds sigv4 support for Elastic Load Balancing, and fixes several other issues including issues making anonymous AWS Security Token Service requests.

4.17.1 Changes

- Add Amazon EC2 Classic Link support (:sha: 5dbd2d7)
- Add query string to body for anon STS POST (issue 2812, commit 6513789)
- Fix bug that prevented initializing a dynamo item from existing item (issue 2764, commit 743e814)
- – switchover-sigv4: Add integ tests for sigv4 switchover Switch elb/ec2 over to signature version 4 (commit 0dadce8)
- Return SetStackPolicyResponse - (issue 2822, issue 2346, issue 2639, commit c4defb4)
- Added ELB Attributes to docs. (issue 2821, commit 5dfeba9)
- Fix bug by using correct string joining syntax. (issue 2817, commit 8426148)
- Fix SES get_identity_dkim_attributes when input length > 1. (issue 2810, commit cc4d42d)
- DynamoDB table batch_get fails to process all remaining results if single batch result is empty. (issue 2809, commit a193bc0)
- Added support for additional fields in EMR objects. (issue 2807, commit 2936ac0)
- Pass version_id in copy if key is versioned. (issue 2803, commit 66b3604)
- Add support for SQS PurgeQueue operation. (issue 2806, commit 90a5d44)
- Update documentation for launchconfig. (issue 2802, commit 0dc8412)
- Remove unimplemented config param. (issue 2801, issue 2572, commit f1a5ebd)
- Add support for private hosted zones. (issue 2785, commit 2e7829b)
- Fix Key.change_storage_class so that it obeys dst_bucket. (issue 2752, commit 55ed184)
- Fix for s3put host specification. (issue 2736, issue 2522, commit 1af31f2)
- Improve handling of Glacier HTTP 204 responses. (issue 2726, commit c314298)
- Fix raising exception syntax in Python 3. (issue 2735, issue 2563, commit 58f76f6)

- Privatezone: Adding unit/integration test coverage (issue 1, commit d1ff14e)
- Minor documentation/pep8 fixes. (issue 2753, commit 6a853be)
- Correct argument type in doc string. (issue 2728, commit 1ddf6df)
- Use exclusive start key to get all items from DynamoDB query. (issue 2676, issue 2573, commit 419d8a5)
- Updated link to current config documentation. (issue 2755, commit 9be3f85)
- Fix the SQS certificate error for region cn-north-1. (issue 2766, commit 1d5368a)
- Adds support for getting health checker IP ranges from Route53. (issue 2792, commit ee14911)
- fix: snap.create_volume documentation lists general purpose ssd. Fixes @2774. (issue 2774, commit 36fae2b)
- Fixed param type in get_contents_to_filename docstring. (issue 2783, commit 478f66a)
- Update DynamoDB local example to include fake access key id. (issue 2791, commit 2c1f8d5)
- Added 'end' attribute to ReservedInstance. (issue 2793, issue 2757, commit 28814d8)
- Parse ClusterStatus's StateChangeReason. (issue 2696, commit 48c5d17)
- Adds SupportedProducts field to EMR JobFlow objects. (issue 2775, commit 6771d04)
- Fix EMR endpoint. (issue 2750, commit 8329e02)
- Detect old-style S3 URL for auto-sigv4. (issue 2773, commit f5be409)
- Throw host warning for cloudsearch domain (issue 2765, commit 9af6f41)
- Fix CloudSearch2 to work with IAM-based search and upload requests (issue 2717, commit 9f4fe8b)
- iam: add support for Account Password Policy APIs (issue 2574, commit 6c9bd53)
- Handle sigv4 non-string header values properly (issue 2744, commit e043e4b)
- Url encode query string for pure query (issue 2720, commit bbbf9d2)

4.18 boto v2.34.0

date 2014/10/23

This release adds region support for `eu-central-1`, support to create virtual mfa devices for Identity and Access Management, and fixes several sigv4 issues.

4.18.1 Changes

- Calculate sha_256 correctly for s3 (issue 2691, commit c0a001f)
- Fix MTurk typo. (issue 2429, issue 2428, commit 9bfff19)
- Fix Amazon Cognito links in docs (issue 2674, commit 7c28577)
- Add the ability to IAM to create a virtual mfa device. (issue 2675, commit 075d402)
- PEP8 tidy up for several modules. (issue 2673, commit 38abbd9)
- Fix s3 create multipart upload for sigv4 (issue 2684, commit fc73641)
- Updated endpoints.json for cloudwatch logs to support more regions. (issue 2685, commit 5db2ea8)

4.19 boto v2.33.0

date 2014/10/08

This release adds support for Amazon Route 53 Domains, Amazon Cognito Identity, Amazon Cognito Sync, the DynamoDB document model feature, and fixes several issues.

4.19.1 Changes

- Added TaggedEC2Object.remove_tags. (issue 2610, issue 2269, issue 2414, commit bce8fcf)
- Fixed 403 error from url encoded User-Agent header (issue 2621, commit 2043a89)
- Inserted break when iterating Route53 records. (issue 2631, commit 2de8716)
- Fix typo in ELB ConnectionSettings attribute (issue 2602, commit 63bd53b)
- PEP8 fixes to various common modules. (issue 2611, commit 44d873d)
- Route Tables: Update describe_route_tables to support additional route types (VPC peering connection, NIC). (issue 2598, issue 2597, commit bbe8ce7)
- Fix an error in Python 3 when creating launch configs. Enables AutoScaling unit tests to run by default. (issue 2591, commit fb4aeeec)
- Use svg instead of png to get better image quality. (issue 2588, commit 1de6b41)
- STS now signs using sigv4. (issue 2627, commit 36b247f)
- Added support for Amazon Cognito. (issue 2608, commit fa3a39e)
- Fix bug where sigv4 custom metadata headers were presigned incorrectly. (issue 2604, commit 8853e8e)
- Add some regions to cloudsearch (issue 2593, commit 8c6ea21)
- fix typo in s3 tutorial (issue 2612, commit 92dd581)
- fix ELB ConnectionSettings values in documentation (issue 2620, commit d2231a2)
- Few typo in docstring (issue 2590, commit 0238747)
- Add support for Amazon Route 53 Domains. (issue 2601, commit d149a87)
- Support EBS encryption in BlockDeviceType. (issue 2587, issue 2480, commit 7a39741)
- Fix a typo in auth.py: Beijing -> Beijing. (issue 2585, commit 8525616)
- Update boto/cacerts/cacerts.txt. (issue 2567, commit 02b836c)
- route53 module: tidy up to meet PEP8 better. (issue 2571, commit 3a3e960)
- Update count_slow documentation. (issue 2569, commit e926d2d)
- iam module: tidy up to meet PEP8 better. (issue 2566, commit 3c83da9)
- Assigning ACL ID to network_acl_id instead of route_table_id. (issue 2548, commit c017b02)
- Avoid infinite loop with bucket listing and encoding_type='url'. (issue 2562, issue 2561, commit 39cbcb5)
- Use urllib timeout param instead of hacking socket global timeout. (issue 2560, issue 1935, commit c1dd1fb)
- Support non-ascii unicode strings in _get_all_query_args. Fixes: #2558, #2559. (issue 2559, issue 2558, commit 069d04b)
- Truncated Response Handling in Route53 ListResourceRecordSets. (issue 2542, commit 3ba380f)

- Update to latest OpsWorks API. (issue 2547, commit ac2b311)
- Better S3 key repr support for unicode. (issue 2525, issue 2516, commit 8198884)
- Skip test when locale is missing. (issue 2554, issue 2540, commit 2b87583)
- Add profile_name support to SQS. (issue 2459, commit 3837951)
- Include test_endpoints.json in source distribution. (issue 2550, commit 7f907b7)
- Pass along params in make_request for elastic transcoder api. (issue 2537, commit 964999e)
- Documents not found behavior of get_item(). (issue 2544, commit 9b9c1c4)
- Support auth when headers contains bytes. (issue 2521, issue 2520, commit 885348d)
- PEP8 style fixes for ElastiCache. (issue 2539, commit bd0d6db)
- PEP8 style fixes for SES. (issue 2538, commit c620c43)
- Doc updates for CloudSearch. (issue 2546, commit 9efebe2)
- Update to latest Redshift API. (issue 2545, commit 9151092)
- Update to latest support API. (issue 2541, issue 2426, commit 8cf1b52)
- Uses file name as archive description when uploading to glacier. (issue 2535, issue 2528, commit 38478c1)
- Fix the ec2.elb.listener.Listener class's __getitem__ method. (issue 2533, commit 7b67f98)
- Add recognized HTTP headers for S3 metadata. (issue 2477, issue 2050, commit c8c625a)
- Fix class name for document. (issue 2530, commit 2f0e689)
- Copy CloudSearch proxy settings to endpoint services. (issue 2513, commit 3cbbc21)
- Merge branch 'develop' into cloudsearch2-proxy (commit 5b424db)
- Add IAMer as an application built on boto. (issue 2515, commit 1f35224)

4.20 boto v2.32.1

date 2014/08/04

This release fixes an incorrect Amazon VPC peering connection call, and fixes several minor issues related to Python 3 support including a regression when pickling authentication information.

4.20.1 Fixes

- Fix bin scripts for Python 3. (issue 2502, issue 2490, commit cb78c52)
- Fix parsing of EMR step summary response. (issue 2456, commit 2ffb00a)
- Update wheel to be universal for py2/py3. (issue 2478, commit e872d94)
- Add pypy to tox config. (issue 2458, commit 16c6fbe)
- Fix Glacier file object hash calculation. (issue 2489, issue 2488, commit a9463c5)
- PEP8 fixes for Glacier. (issue 2469, commit 0575a54)
- Use ConfigParser for Python 3 and SafeConfigParser for Python 2. (issue 2498, issue 2497, commit f580f73)
- Remove redundant __future__ imports. (issue 2496, commit e59e199)

- Fix dynamodb.types.Binary non-ASCII handling. (issue 2492, issue 2491, commit 16284ea)
- Add missing dependency to requirements.txt. (issue 2494, commit 33db71a)
- Fix TypeError when getting instance metadata under Python 3. (issue 2486, issue 2485, commit 6ff525e)
- Handle Cloudsearch indexing errors. (issue 2370, commit 494a091)
- Remove obsolete md5 import routine. (issue 2468, commit 9808a77)
- Use encodebytes instead of encodestring. (issue 2484, issue 2483, commit 984c5ff)
- Fix an auth class pickling bug. (issue 2479, commit 07d6424)

4.21 boto v2.32.0

date 2014/07/30

This release includes backward-compatible support for Python 3.3 and 3.4, support for IPv6, Amazon VPC connection peering, Amazon SNS message attributes, new regions for Amazon Kinesis, and several fixes.

4.21.1 Python 3 Support

- DynamoDB (issue 2441, commit 0ef0466, issue 2473, commit 102c3b6, issue 2453)
- CloudWatch Logs (issue 2448, commit 23cbcd1)
- Support (issue 2406, commit 7b489a0)
- Elastic Beanstalk (issue 2372, commit d45d00e)
- CloudSearch (issue 2439, commit 25416f9, issue 2432, commit b17f2d9)
- STS (issue 2435, commit 1c1239b)
- SimpleDB (issue 2403, commit 604318d)
- EC2 (issue 2424, commit 5e5dc4c)
- VPC (issue 2399, commit 356da91)
- OpsWorks (issue 2402, commit 68d15a5)
- CloudWatch (issue 2400, commit a4d0a7a)
- SWF (issue 2397, commit 6db918e)
- MWS (issue 2385, commit 5347fbd)
- ELB (issue 2384, commit 4dcc9be)
- Elastic Transcoder (issue 2382, commit 40c5e35)
- EMR (issue 2381, commit edf4020)
- Route53 (issue 2359, commit 15514f7)
- Glacier (issue 2357, commit a41042e)
- RedShift (issue 2362, commit b8888cc)
- CloudFront (issue 2355, commit f2f54b1)
- ECS (issue 2364, commit ab84969)

- Fix pylintrc to run with pylint/python 3. (issue 2366, commit 6292ab2)
- SNS (issue 2365, commit 170f735)
- AutoScaling (issue 2393, commit 6a78057)
- Direct Connect (issue 2361, commit 8488d94)
- CloudFormation (issue 2373, commit 9872f27)
- IAM (issue 2358, commit 29ad3e3)
- ElastiCache (issue 2356, commit 2880f91)
- SES (issue 2354, commit 1db129e)
- Fix S3 integration test on Py3. (issue 2466, commit f3eb4cd)
- Use unittest.mock if exists. (issue 2451, commit cc58978)
- Add tests/compat.py for test-only imports. (issue 2442, commit 556f3cf)
- Add backward-compatible support for Python 3.3+ (S3, SQS, Kinesis, CloudTrail). (issue 2344, issue 677, commit b503f4b)

4.21.2 Features

- Add marker param to describe all ELBs. (issue 2433, commit 49af8b6)
- Update .travis.yml to add pypy. (issue 2440, commit 4b8667c)
- Add ‘include_all_instances’ support to ‘get_all_instance_status’. (issue 2446, issue 2230, commit 5949012)
- Support security tokens in configuration file profiles. (issue 2445, commit a16bcfd)
- Singapore, Sydney and Tokyo are missing in Kinesis Region. (issue 2434, commit 723290d)
- Add support for VPC connection peering. (issue 2438, commit 63c78a8)
- Add separate doc requirements. (issue 2412, commit 2922d89)
- Route53 support IP health checks (issue 2195, commit 319d44e)
- IPv6 support when making connections (issue 2380, commit 1e70179)
- Support SNS message attributes (issue 2360, commit ec106bd)
- Add “attributes” argument to boto.dynamodb2.table.Table.batch_get. (issue 2276, commit fe67f43)
- Add documentation for top-level S3 module. (issue 2379, commit db77546)

4.21.3 Fixes

- Prevent an infinite loop. (issue 2465, commit 71b795a)
- Updated documentation for copy_image. (issue 2471, commit f9f683a)
- Fixed #2464 added keyword “detailed” to docs. (issue 2467, issue 2464, commit eb26fde)
- Retry installation commands on Travis CI. (issue 2457, commit a9e8057)
- Fix for run_instances() network_interfaces argument documentation. (issue 2461, commit 798fd70)
- pyami module: tidy up to meet PEP8 better. (issue 2460, commit e5a23ed)
- Updating documentation on cloudsearch regions. (issue 2455, commit de284a4)

- Fixing lost errors bug in cloudsearch2 commit implementation. (issue 2408, commit fedb937)
- Import json from boto.compat for several modules. (issue 2450, commit 55e716b)
- Relocate MWS requirements checks; closes #2304, #2314. (issue 2314, issue 2304, commit 6a8f98b)
- Added support for creating EMR clusters with a ServiceRole. (issue 2389, commit 7693956)
- Doc fix: doc_service instead of service on Deleting. (issue 2419, commit f7b7980)
- Fix dummy value typo on aws_access_key_id. (issue 2418, commit fc2a212)
- Fix typo; add test. (issue 2447, commit effa8a8)
- Fix CloudWatch Logs docstring. (issue 2444, commit d4a2b02)
- Fix S3 mock encoding bug (issue 2443, commit 8dca89b)
- Skip the ETag header check in response while using SSE-C encryption of S3. (issue 2368, commit 907fc6d)
- Fix Beanstalk exception handling. (issue 2431, commit 40f4b5d)
- EC2 UserData encoding fix (Full version of #1698). (issue 2396, issue 1698, commit 78300f1)
- Fetch S3 key storage class on-demand. (issue 2404, commit 8c4cc67)
- Added documentation for /manage/cmdshell.py. (issue 2395, commit 5a28d1c)
- Remove redundant lines in auth.py. (issue 2374, commit 317e322)
- Fix SWF continue_as_new_workflow_execution start_to_close_timeout. (issue 2378, commit 5101b06)
- Fix StringIO imports and invocations. (issue 2390, commit 03952c7)
- Fixed wrong call of urlparse. (issue 2387, commit 4935f67)
- Update documentation on Valid Values for ses:SetIdentityNotificationTopic. (issue 2367, commit 3f5de0d)
- Correct list_saml_providers to return all items. (issue 2338, commit 9e9427f)
- Fixing ELB unit tests. Also did some PEP8 cleanup on ELB code. (issue 2352, commit 5220621)
- Documentation updates. (issue 2353, commit c9233d4)

4.22 boto v2.31.1

date 2014/07/10

This release fixes an installation bug in the 2.31.0 release.

4.23 boto v2.31.0

date 2014/07/10

This release adds support for Amazon CloudWatch Logs.

4.23.1 Changes

- Add support for Amazon CloudWatch Logs. (commit 125c94d)

4.24 boto v2.30.0

date 2014/07/01

This release adds new Amazon EC2 instance types, new regions for AWS CloudTrail and Amazon Kinesis, Amazon S3 presigning using signature version 4, and several documentation and bugfixes.

4.24.1 Changes

- Add EC2 T2 instance types ([commit 544f8925cb](#))
- Add new regions for CloudTrail and Kinesis ([commit 4d67e19914](#))
- Fixed some code formatting and typo in SQS tutorial docs. ([issue 2332](#), [commit 08c8fed](#))
- Documentation update – Child workflows and poll API. ([issue 2333](#), [issue 2063](#), [issue 2064](#), [commit 4835676](#))
- DOC Tutorial update for metrics and use of dimensions property. ([issue 2340](#), [issue 2336](#), [commit 45fda90](#))
- Let people know only EC2 supported for cloudwatch. ([issue 2341](#), [commit 98f03e2](#))
- Add namespace to AccessControlPolicy xml representation. ([issue 2342](#), [commit ce07446](#))
- Make ip_addr optional in Route53 HealthCheck. ([issue 2345](#), [commit 79c35ca](#))
- Add S3 SigV4 Presigning. ([issue 2349](#), [commit 125c4ce](#))
- Add missing route53 autodoc. ([issue 2343](#), [commit 6472811](#))
- Adds scan_index_forward and limit to DynamoDB table query count. ([issue 2184](#), [commit 4b6d222](#))
- Add method TaggedEC2Object.add_tags(). ([issue 2259](#), [commit eea5467](#))
- Add network interface lookup to EC2. Add update/attach/detach methods to NetworkInterface object. ([issue 2311](#), [commit 4d44530](#))
- Parse date/time in a locale independent manner. ([issue 2317](#), [issue 2271](#), [commit 3b715e5](#))
- Add documentation for delete_hosted_zone. ([issue 2316](#), [commit a0fdd39](#))
- s/existance/existence/ ([issue 2315](#), [commit b8dfa1c](#))
- Add multipart upload section to the S3 tutorial. ([issue 2308](#), [commit 99953d4](#))
- Only attempt shared creds load if path is a file. ([issue 2305](#), [commit 0bffa3b](#))

4.25 boto v2.29.1

date 2014/05/30

This release fixes a critical bug when the provider is not set to `aws`, e.g. for Google Storage. It also fixes a problem with connection pooling in Amazon CloudSearch.

4.25.1 Changes

- Fix crash when provider is google. ([issue 2302](#), [commit 33329d5888](#))
- Fix connection pooling issue with CloudSearch ([commit 82e83be12a](#))

4.26 boto v2.29.0

date 2014/05/29

This release adds support for the AWS shared credentials file, adds support for Amazon Elastic Block Store (EBS) encryption, and contains a handful of fixes for Amazon EC2, AWS CloudFormation, AWS CloudWatch, AWS CloudTrail, Amazon DynamoDB and Amazon Relational Database Service (RDS). It also includes fixes for Python wheel support.

A bug has been fixed such that a new exception is thrown when a profile name is explicitly passed either via code (`profile="foo"`) or an environment variable (`AWS_PROFILE=foo`) and that profile does not exist in any configuration file. Previously this was silently ignored, and the default credentials would be used without informing the user.

4.26.1 Changes

- Added support for shared credentials file. (issue 2292, commit d5ed49f)
- Added support for EBS encryption. (issue 2282, commit d85a449)
- Added GovCloud CloudFormation endpoint. (issue 2297, commit 0f75fb9)
- Added new CloudTrail endpoints to endpoints.json. (issue 2269, commit 1168580)
- Added 'name' param to documentation of ELB LoadBalancer. (issue 2291, commit 86e1174)
- Fix typo in ELB docs. (issue 2294, commit 37aaa0f)
- Fix typo in ELB tutorial. (issue 2290, commit 40a758a)
- Fix OpsWorks `connect_to_region` exception. (issue 2288, commit 26729c7)
- Fix timezones in CloudWatch date range example. (issue 2285, commit 138a6d0)
- Fix description of param tags into `rds2.create_db_subnet_group`. (issue 2279, commit dc1037f)
- Fix the incorrect name of a test case. (issue 2273, commit ee195a1)
- Fix "consistent" argument to `boto.dynamodb2.table.Table.batch_get`. (issue 2272, commit c432b09)
- Update the wheel to be python 2 compatible only. (issue 2286, commit 6ad0b75)
- Crate.io is no longer a package index. (issue 2289, commit 7f23de0)

4.27 boto v2.28.0

date 2014/05/08

This release adds support for Amazon SQS message attributes, Amazon DynamoDB query filters and enhanced conditional operators, adds support for the new Amazon CloudSearch 2013-01-01 API and includes various features and fixes for Amazon Route 53, Amazon EC2, Amazon Elastic Beanstalk, Amazon Glacier, AWS Identity and Access Management (IAM), Amazon S3, Mechanical Turk and MWS.

4.27.1 Changes

- Add support for SQS message attributes. (issue 2257, commit a04ca92)

- Update DynamoDB to support query filters. (issue 2242, commit 141eb71)
- Implement new Cloudsearch API 2013-01-01 as cloudsearch2 module (commit b0ababa)
- Miscellaneous improvements to the MTurk CLI. (issue 2188, commit c213ff1)
- Update MWS to latest API version and adds missing API calls. (issue 2203, issue 2201, commit 8adf720, commit 8d0a6a8)
- Update EC2 `register_image` to expose an option which sets whether an instance store is deleted on termination. The default value is left as-is. (commit d295ee9)
- Correct typo “possile” → “possible”. (issue 2196, commit d228352)
- Update Boto configuration tutorial (issue 2191, commit f2a7a08)
- Clarify that MTurkConnection.get_assignments attributes are actually strings. (issue 2187, issue 2176, commit 075636b)
- Fix EC2 documentation typo (issue 2178, commit 2627843)
- Add support for ELB Connection Draining attribute. (issue 2174, issue 2173, commit 78fa43c)
- Add support for setting failure threshold for Route53 health checks. (issue 2171, issue 2170, commit 15b812f)
- Fix specification of Elastic Beanstalk tier parameter. (issue 2168, commit 4492e86)
- Fixed part of roboto for euca2ools. (issue 2166, issue 1730, commit 63b7a34)
- Fixed removing policies from listeners. (issue 2165, issue 1708, commit e5a2d9b)
- Reintroduced the `reverse` fix for DDB. (issue 2163, commit 70ec722)
- Several fixes to DynamoDB describe calls. (issue 2161, issue 1649, issue 1663, commit 84fb748)
- Fixed how `reverse` works in DynamoDBv2. (issue 2160, issue 2070, issue 2115, commit afdd805)
- Update Kinesis exceptions (issue 2159, issue 2153, commit 22c6751)
- Fix ECS problem using new-style classes (issue 2103, commit dc466c7)
- Add support for passing region info from SWF layer2 to layer1 (issue 2137, commit 0dc8ce6)
- Handle plus signs in S3 metadata (issue 2145, commit c2a0f95)
- Fix Glacier vault date parsing (issue 2158, commit 9e7b132)
- Documentation fix. (issue 2156, commit 7592a58)
- Fix Route53 evaluate target health bug. (issue 2157, commit 398bb62)
- Removing obsolete core directory. (issue 1987, commit 8e83292)
- Improve IAM behavior in the cn-north-1 region. (issue 2152, commit 4050e70)
- Add SetIdentityFeedbackForwardingEnabled and SetIdentityNotificationTopic for SES. (issue 2130, issue 2128, commit 83002d5)
- Altered Route53 bin script to use UPSERT rather than CREATE. (issue 2151, commit 2cd20e7)

4.28 boto v2.27.0

date 2014/03/06

This release adds support for configuring access logs on Elastic Load Balancing (including what Amazon Simple Storage Service (S3) bucket to use & how frequently logs should be added to the bucket), adds request hook documentation & a host of doc updates/bugfixes.

4.28.1 Changes

- Added support for `AccessLog` in ELB (issue 2150, commit 7aa35ea)
- Added better `BlockDeviceType` deserialization in Autoscaling. (issue 2149, commit 04d29a5)
- Updated CloudFormation documentation (issue 2147, commit 2535aca)
- Updated Kinesis documentation (issue 2146, commit 01425dc)
- Add optional bucket tags to `Iss3` output. (issue 2132, commit 0f35924)
- Fix getting instance types for Eucalyptus 4.0. (issue 2118, commit 18dc07d)
- Fixed how quoted strings are handled in SigV4 (issue 2142, commit 2467547)
- Use system supplied certs without a bundle file (issue 2139, commit 70d15b8)
- Fixed incorrect test failures in EC2 `trim_snapshots` (commit 1fa9df7)
- Raise any exceptions that are tagSet not found (commit 56d7d3e)
- Added request hook docs (issue 2129, commit 64eedce)
- Fixed Route53 `alias-healthcheck` (issue 2126, commit 141077f)
- Fixed Elastic IP association in EC2 (issue 2131, issue 1310, commit d75fdfa)
- Fixed builds on Travis for installing dependencies (commit 5e84e30)
- Support printing tags on buckets when listing buckets (commit c42a5dd)
- PEP8/pyflakes/(some)pylint (commit 149175e)

4.29 boto v2.26.1

date 2014/03/03

This release fixes an issue with the newly-added `boto.rds2` module when trying to use `boto.connect_rds2`. Parameters were not being passed correctly, which would cause an immediate error.

4.29.1 Changes

- Fixed `boto.connect_rds2` to use `kwargs`. (commit 3828ece)

4.30 boto v2.26.0

date 2014/02/27

This release adds support for MFA tokens in the AWS STS `assume_role` & the introduction of the `boto.rds2` module (which has full support for the entire RDS API). It also includes the addition of request hooks & many bugfixes.

4.30.1 Changes

- Added support for MFA in STS AssumeRole. (commit 899810c)
- Fixed how DynamoDB v2 works with Global Secondary Indexes. (issue 2122, commit f602c95)
- Add request hooks and request logger. (issue 2125, commit e8b20fe)
- Don't pull the security token from the environment or config when a caller supplies the access key and secret. (issue 2123, commit 4df1694)
- Read EvaluateTargetHealth from Route53 resource record set. (issue 2120, commit 0a97158)
- Prevent implicit string decode in hmac-v4 handlers. (issue 2037, issue 2033, commit 8e56a5f)
- Updated Datapipeline to include all current regions. (issue 2121, commit dff5e3e)
- Bug fix for Google Storage generate_url authentication. (issue 2116, issue 2108, commit 5a50932)
- Handle JSON error responses in BotoServerError. (issue 2113, issue 2077, commit 221085e)
- Corrected a typo in SQS tutorial. (issue 2114, commit 7ed41f7)
- Add CloudFormation template capabilities support. (issue 2111, issue 2075, commit 65a4323)
- Add SWF layer1_decisions to docs. (issue 2110, issue 2062, commit 6039cc9)
- Add support for request intervals in health checks. (issue 2109, commit 660b01a)
- Added checks for invalid regions to the bin scripts (issue 2107, commit bbb9f1e)
- Better error output for unknown region - (issue 2041, issue 1983, commit cd63f92)
- Added certificate tests for CloudTrail. (issue 2106, commit a7e9b4c)
- Updated Kinesis endpoints. (commit 7bd4b6e)
- Finished implementation of RDS's DescribeDBLogFiles. (issue 2084, commit f3c706c)
- Added support for RDS log file downloading. (issue 2086, issue 1993, commit 4c51841)
- Added some unit tests for CloudFront. (issue 2076, commit 6c46b1d)
- GS should ignore restore_headers as they are never set. (issue 2067, commit f02aeb3)
- Update CloudFormation to support the latest API. (issue 2101, commit ea1b1b6)
- Added Route53 health checks. (issue 2054, commit 9028f7d)
- Merge branch 'rds2' into develop Fixes #2097. (issue 2097, commit 6843c16)
- Fix Param class convert method (issue 2094, commit 5cd4598)
- Added support for Route53 aliasing. (issue 2096, commit df5fa40)
- Removed the dependence on example.com within the Route53 tests. (issue 2098, commit 6ce9e0f)
- Fixed has_item support in DynamoDB v2. (issue 2090, commit aada5d3)
- Fix a little typo bug in the S3 tutorial. (issue 2088, commit c091d27)

4.31 boto v2.25.0

date 2014/02/07

This release includes Amazon Route53 service and documentation updates, preliminary log file support for Amazon Relational Database Service (RDS), as well as various other small fixes. Also included is an opt-in to use signature version 4 with Amazon EC2.

IMPORTANT - This release also include a **SIGNIFICANT** underlying change to the Amazon S3 `get_bucket` method, to addresses the blog post by [AppNeta](#). We've altered the default behavior to now perform a HEAD on the bucket, in place of the old GET behavior (which would fetch a zero-length list of keys).

This should reduce all users costs & should also be *mostly* backward-compatible. **HOWEVER**, if you were previously parsing the exception message from `S3Connection.get_bucket`, you *will* have to change your code (see the S3 tutorial for details). HEAD does *not* return as detailed of error messages & while we've attempted to patch over as much of the differences as we can, there may still be edge-cases over the prior behavior.

4.31.1 Features

- Add support for Route53 API version 2013-04-01 ([issue 2080](#), [commit 600dcd0](#))
- Add option to opt-in for EC2 SigV4 ([issue 2074](#), [commit 4d780bd](#))
- Add Autoscale feature to get all adjustment types ([issue 2058](#), [issue 1538](#), [commit b9c7e15](#))
- Add Route53 unit tests ([issue 2066](#), [commit e859576](#))
- Add a basic Route53 tutorial ([issue 2060](#), [commit f0ad46b](#))
- Add Autoscale associated public IP to launch configuration ([issue 2051](#), [issue 2028](#), [issue 2029](#), [commit c58bda6](#))
- Add option to pass VPC zone identifiers as a Python list ([issue 2047](#), [issue 1772](#), [commit 07ef9e1](#))
- Add RDS call to get all log files ([issue 2040](#), [issue 1994](#), [commit 925b8cb](#))

4.31.2 Bugfixes

- Changed S3 `get_bucket` to use HEAD in place of GET. ([issue 2078](#), [issue 2082](#), [commit 016be83](#))
- Fix EMR's `describe_cluster_command`. ([issue 2034](#), [commit 1c5621e](#))
- Tutorial small code fix ([issue 2072](#), [commit 38e7db1](#))
- Fix CloudFront string representation ([issue 2069](#), [commit 885c397](#))
- Route53 doc cleanup ([issue 2059](#), [commit d2fc38e](#))
- Fix MWS parsing of `GetProductCategoriesForASIN` response. ([issue 2024](#), [commit 0af08ce](#))
- Fix SQS docs for `get_queue_attributes` ([issue 2061](#), [commit 1cdc326](#))
- Don't insert a '?' in URLs unless there is a query string ([issue 2042](#), [issue 1943](#), [commit c15ce60](#))

4.32 boto v2.24.0

date 2014/01/29

This release adds M3 instance types to Amazon EC2, adds support for dead letter queues to Amazon Simple Queue Service (SQS), adds a single JSON file for all region and endpoint information and provides several fixes to a handful of services and documentation. Additionally, the SDK now supports using AWS Signature Version 4 with Amazon S3.

4.32.1 Features

- Load region and endpoint information from a JSON file (commit b9dbaad)
- Return the x-amz-restore header with GET KEY and fix provider prefix. (issue 1990, commit 43e8e0a)
- Make S3 key validation optional with the `validate` parameter (issue 2013, issue 1996, commit fd6b632)
- Adding new eu-west-1 and eu-west-2 endpoints for SES. (issue 2015, commit d5ef862, commit 56ba3e5)
- Google Storage now uses new-style Python classes (issue 1927, commit 86c9f77)
- Add support for step summary list to Elastic MapReduce (issue 2011, commit d3af158)
- Added the M3 instance types. (issue 2012, commit 7c82f57)
- Add credential profile configuration (issue 1979, commit e3ab708)
- Add support for dead letter queues to SQS (commit 93c7d05)

4.32.2 Bugfixes

- Make the Lifecycle Id optional and fix prefix=None in XML generation. (issue 2021, commit 362a04a)
- Fix DynamoDB query limit bug (issue 2014, commit 7ecb3f7)
- Add documentation about the `version_id` behavior of Key objects. (issue 2026, commit b6b242c)
- Fixed typo in Table.create example (issue 2023, commit d81a660)
- Adding a license/copyright header. (issue 2025, commit 26ded39)
- Update the docstring for the SNS subscribe method (issue 2017, commit 4c806de)
- Renamed unit test with duplicate name (issue 2016, commit c7bd0bd)
- Use UTC instead of local time in `test_refresh_credentials` (issue 2020, commit b5a2eaf)
- Fix missing `security_token` option in some connection classes (issue 1989, issue 1942, commit 2b72f32)
- Fix listing S3 multipart uploads with some parameter combinations (issue 2000, commit 49045bc)
- Fix `elbadmin` crash because of non-existent instances in load balancer (issue 2001, commit d47cc14)
- Fix anonymous S3 fetch test case (issue 1988, issue 1992, commit 8fb1666)
- Fix `elbadmin` boto import (issue 2002, commit 674c3a6)
- Fixing SQS tutorial to correctly describe behavior of the write operation (issue 1986, commit 6147d86)
- Fix various grammar mistakes (issue 1980, commit ada40b5)

4.33 boto v2.23.0

date 2014/01/10

This release adds new pagination & date range filtering to Amazon Glacier, more support for selecting specific attributes within Amazon DynamoDB, security tokens from environment/config variables & many bugfixes/small improvements.

4.33.1 Features

- Added pagination & date range filtering to Glacier inventory options. (issue 1977, commit 402a305)
- Added the ability to select the specific attributes to fetch in the `scan` & `get_item` calls within DynamoDB v2. (issue 1945, issue 1972, commit f6451fb & commit 96cd413)
- Allow getting a security token from either an environment or configuration variable. (:issue:, :sha:)
- Ported the `has_item` call from the original DynamoDB (v1) module to DynamoDB v2. (issue 1973, issue 1822, commit f96e9e3)
- Added an `associate_address_object` method to EC2. (issue 1967, issue 1874, issue 1893, commit dd6180c)
- Added a `download_to_fileobj` method to Glacier, similar to the S3 call of the same name. (issue 1960, issue 1941, commit 67266e5)
- Added support for arbitrary dict inputs to MWS. (issue 1966, commit 46f193f)

4.33.2 Bugfixes

- Made the usage of `is/is not` more consistent. (issue 1930, commit 8597c54)
- Imported `with_statement` for old Python versions (issue 1975, commit a53a574)
- Changed the `Binary` data object within DynamoDB to throw an error if an invalid data type is used. (issue 1963, issue 1956, commit e5d30c8)
- Altered the integration tests to avoid connection errors to certain regions. (commit 2555b8a)
- Changed the GCS resumable upload handler to save tracker files with protection 0600. (commit 7cb344c)
- Documentation:
 - Clarified documentation around the `list_metrics` call in CloudFormation. (issue 1962, commit c996a72)
 - Added `Tag` to the Autoscale API docs. (issue 1964, commit 31118d9)
 - Updated the AWS Support documentation to the latest. (commit 29f9264)

4.34 boto v2.22.1

date 2014/01/06

This release fixes working with keys with special characters in them while using Signature V4 with Amazon Simple Storage Service (S3). It also fixes a regression in the `ResultSet` object, re-adding the `nextToken` attribute. This was most visible from within Amazon Elastic Compute Cloud (EC2) when calling the `get_spot_price_history` method.

Users in the `cn-north-1` region or who make active use of `get_spot_price_history` are recommended to upgrade.

4.34.1 Bugfixes

- Fixed key names with special characters in S3 when using SigV4. (commit 8b37180)
- Re-added the `nextToken` attribute to the EC2 result set object. (issue 1968, commit 6928928)

4.35 boto v2.22.0

date 2014/01/02

This release updates the Auto Scaling to support the latest API, the ability to control the response sizes in Amazon DynamoDB queries/scans & a number of bugfixes as well.

4.35.1 Features

- Updated Auto Scaling to support the latest API. (commit 9984c4f)
- Added the ability to alter response sizes in DynamoDB queries/scans. (issue 1949, commit 6761b01)

4.35.2 Bugfixes

- Fix string instance tests. (issue 1959, commit ee203bf)
- Add missing parameters to `get_spot_price_history` method. (issue 1958, commit f635474)
- Fix unicode string parameter handling in `S3Connection`. (issue 1954, issue 1952, commit 12e6b0c)
- Fix typo in docstring for `SSHClient.run`. (issue 1953, commit 5263b20)
- Properly handle getopt long options in `s3put`. (issue 1950, issue 1946, commit cf693ff)

4.36 boto v2.21.2

date 2013/12/24

This release is a bugfix release which corrects one more bug in the Mechanical Turk objects.

4.36.1 Bugfixes

- Fixed a missed inheritance bug in `mturk`. (issue 1936, commit 0137f29)

4.37 boto v2.21.1

date 2013/12/23

This release is a bugfix release which corrects how the Mechanical Turk objects work & a threading issue when using `datetime.strptime`.

4.37.1 Bugfixes

- Added `cn-north-1` to regions. (commit 9c89de1)
- Fixed threading issues related to `datetime.strptime`. (issue 1898, commit 2ef66c9)
- Updated all the old-style inheritance calls. (issue 1918, issue 1936, issue 1937, commit 39a997f & commit 607624f)
- Documentation:

- Added missed notes about the cn-north-1 region. (commit 738c8cb)
- Added the C3 family of EC2 instances. (issue 1938, commit 05b7482)

4.38 boto v2.21.0

date 2013/12/19

This release adds support for the latest AWS OpsWorks, AWS Elastic Beanstalk, Amazon DynamoDB, Amazon Elastic MapReduce (EMR), Amazon Simple Storage Service (S3), Amazon Elastic Transcoder, AWS CloudTrail, and AWS Support APIs. It also includes documentation and other fixes.

Note: Although Boto now includes support for the newly announced China (Beijing) Region, the service endpoints will not be accessible until the Region's limited preview is launched in early 2014. To find out more about the new Region and request a limited preview account, please visit <http://www.amazonaws.cn/>.

4.38.1 Features

- Add support for Elastic Transcoder pagination and new codecs (commit dcb1c5a)
- Add support for new CloudTrail calling format (commit aeafe9b)
- Update to the latest Support API (commit 45e1884)
- Add support for arbitrarily large SQS messages stored in S3 via BigMessage. (issue 1917, commit e6cd665)
- Add support for `encoding_type` to S3 (commit 6b2d967)
- Add support for Elastic MapReduce tags (issue 1928, issue 1920, commit b9749c6, commit 8e4c595)
- Add high level support for global secondary indexes in DynamoDB (issue 1924, issue 1913, commit 32dac5b)
- Add support for Elastic Beanstalk worker environments. (issue 1911, commit bbd4fbf)
- Add support for OpsWorks IAM user permissions per stack (commit ac6e4e7)
- Add support for SigV4 to S3 (commit deb9e18)
- Add support for SigV4 to EC2 (commit bdebfe0)
- Add support for SigV4 to ElastiCache (commit b892b45)

4.38.2 Bugfixes

- Add documentation describing account usage for multipart uploads in S3 (commit af03d8d)
- Update `DesiredCapacity` if `AutoScalingGroup.desired_capacity` is not `None`. (issue 1906, issue 1906, issue 1757, commit b6670ce)
- Documentation: add Kinesis API reference (issue 1921, commit c169836)
- Documentation: `sriovNetSupport` instance attribute (issue 1915, commit e1bafce)
- Update RDS documentation for API version: 2013-09-09 (issue 1914, commit fcf702a)
- Switch all classes to new style classes which results in memory use improvements (commit ca36fa2)

4.39 boto v2.20.1

date 2013/12/13

This release fixes an important Amazon EC2 bug related to fetching security credentials via the meta-data service. It is recommended that users of boto-2.20.0 upgrade to boto-2.20.1.

4.39.1 Bugfixes

- Bug fix for IAM security credentials metadata URL. (issue 1912, issue 1908, issue 1907, commit f82e7a5)

4.40 boto v2.20.0

date 2013/12/12

This release adds support for Amazon Kinesis and AWS Direct Connect. Amazon EC2 gets support for new i2 instance types and is more resilient against metadata failures, Amazon DynamoDB gets support for global secondary indexes and Amazon Relational Database Service (RDS) supports new DBInstance and DBSnapshot attributes. There are several other fixes for various services, including updated support for CloudStack and Eucalyptus.

4.40.1 Features

- Add support for Amazon Kinesis (commit d0b684e)
- Add support for i2 instance types to EC2. (commit 0f5371f)
- Add support for DynamoDB Global Secondary Indexes (commit 297cacb)
- Add support for AWS Direct Connect. (issue 1894, issue 1894, commit 3cbca26)
- Add option for sorting SDB dumps to sdbadmin. (issue 1888, issue 1888, commit 070e4f6)
- Add a retry when EC2 metadata is returned as corrupt JSON. (issue 1883, issue 1883, issue 1868, commit 41470a0)
- Added some missing attributes to DBInstance and DBSnapshot. (issue 1880, issue 1880, commit 2751dff)

4.40.2 Bugfixes

- Implement nonzero for DynamoDB Item to consider empty items falsey (issue 1899, commit 808e550)
- Remove *dimensions* from Metric.query() docstring. (issue 1901, issue 1901, commit ba6b8c7)
- Make trailing slashes for EC2 metadata URLs explicit & remove them from userdata requests. This fixes using boto for CloudStack (issue 1900, issue 1900, issue 1897, issue 1856, commit 5f4506e)
- Fix the DynamoDB 'scan in' filter to compare the same attribute types in a list rather than using an attribute set. (issue 1896, issue 1896, commit 5fc59d6)
- Updating Amazon ElastiCache parameters to be optional when creating a new cache cluster. (issue 1876, issue 1876, commit 342b8df)
- Fix honor cooldown AutoScaling parameter serialization to prevent an exception and bad request. (issue 1895, issue 1895, issue 1892, commit fc4674f)

- Fix ignored RDS backup_retention_period when value was 0. (issue 1887, issue 1887, issue 1886, commit a19eb14)
- Use auth_handler to specify host header value including custom ports if possible, which are used by Eucalyptus. (issue 1862, issue 1862, commit ce6df03)
- Fix documentation of launch config in Autoscaling Group. (issue 1881, issue 1881, commit 6f704d9)
- typo: AIM -> IAM (issue 1882, commit 7ea2d5c)

4.41 boto v2.19.0

date 2013/11/27

This release adds support for max result limits for Amazon EC2 calls, adds support for Amazon RDS database snapshot copies and fixes links to the changelog.

4.41.1 Features

- **Add max results parameters to EC2 describe instances and describe tags.** (issue 1873, issue 1873, commit ad8a64a)
- **Add support for RDS CopyDBSnapshot.** (issue 1872, issue 1872, issue 1865, commit bffb758)

4.41.2 Bugfixes

- **Update README.rst to link to ReadTheDocs changelogs.** (issue 1869, commit 26f3dbe)
- **Delete the old changelog in favor of the README link to ReadTheDocs changelogs.** (issue 1870, issue 1870, commit 32bc333)

4.42 boto v2.18.0

date 2013/11/22

This release adds support for new AWS Identity and Access Management (IAM), AWS Security Token Service (STS), Elastic Load Balancing (ELB), Amazon Elastic Compute Cloud (EC2), Amazon Relational Database Service (RDS), and Amazon Elastic Transcoder APIs and parameters. Amazon Redshift SNS notifications are now supported. CloudWatch is updated to use signature version four, issues encoding HTTP headers are fixed and several services received documentation fixes.

4.42.1 Features

- **Add support for new STS and IAM calls related to SAML.** (issue 1867, issue 1867, commit 1c51d17)
- Add SigV4 support to Cloudwatch (commit ef43035)
- **Add support for ELB Attributes and Cross Zone Balancing.** (issue 1852, issue 1852, commit 76f8b7f)
- **Add RDS promote and rename support.** (issue 1857, issue 1857, commit 0b62c70)
- **Update EC2 get_all_snapshots and add support for registering an image with a snapshot.** (issue 1850, issue 1850, commit 3007956)

4.42.2 Bugfixes

- **Fix issues related to encoding of values in HTTP headers when using** unicode. (issue 1864, issue 1864, issue 1839, issue 1829, issue 1828, issue 702, commit 5610dd7)
- **Fix order of Beanstalk documetation to match param order.** (issue 1863, issue 1863, commit a3a29f8)
- **Make sure file is closed before attempting to delete it when downloading** an S3 key. (issue 1791, commit 0e6dcbe)
- **Fix minor CloudTrail documentation typos.** (issue 1861, issue 1861, commit 256a115)
- **Fix DynamoDBv2 tutorial sentence with missing verb.** (issue 1859, issue 1825, issue 1859, commit 0fd5300)
- Fix parameter validation for gs (issue 1858, commit 6b9a869)

4.43 boto v2.17.0

date 2013/11/14

This release adds support for the new AWS CloudTrail service, support for Amazon Redshift's new features related encryption, audit logging, data load from external hosts, WLM configuration, database distribution styles and functions, as well as cross region snapshot copying.

4.43.1 Features

- Add support for AWS CloudTrail (commit 53ba0c9)
- Add support for new Amazon Redshift features (commit d94b48c)

4.43.2 Bugfixes

- Add missing argument for Google Storage resumable uploads. (commit b777b62)

4.44 boto v2.16.0

date 2013/11/08

This release adds new Amazon Elastic MapReduce functionality, provides updates and fixes for Amazon EC2, Amazon VPC, Amazon DynamoDB, Amazon SQS, Amazon Elastic MapReduce, and documentation updates for several services.

4.44.1 Features

- Added recipe for parallel execution of activities to SWF tutorial. (issue 1800, issue 1800, commit 52c5432)
- Added launch_config's parameter associate_ip_address for VPC. (issue 1799, issue 1799, commit 6685adb)
- Update elbadmin add/remove commands to support multiple instance arguments. (issue 1806, issue 1806, commit 4aad26d)
- Added documentation for valid auto scaling event types and tags. (issue 1807, issue 1807, commit 664f6e8)

- Support VPC tenancy restrictions and filters for DHCP options. (issue 1801, issue 1801, commit 8c5d8de)
- Add VPC network ACL support. (issue 1809, issue 1098, issue 1809, commit 9043d09)
- Add convenience functions to make DynamoDB2 behave more like DynamoDB (issue 1780, commit 2ceca)
- EC2 `cancel_spot_instance_requests` now returns a list of `SpotInstanceRequest` objects. (issue 1811, issue 1811, issue 1754, commit f3361b9)
- Fix VPC `DescribeVpnConnections` call argument; Add support for `static_routes_only` when creating a new VPC. (issue 1816, issue 1816, issue 1481, commit b408637)
- Add a section about DynamoDB Local to the DynamoDBv2 high level docs. (issue 1821, issue 1821, issue 1818, commit 639505f)
- Add support for new Elastic MapReduce APIs (issue 1836, commit 5562264)
- Modify EMR `add_jobflow_steps` to return a `JobFlowStepList`. (issue 1838, issue 1838, commit ef9564f)
- Generate docs for `route53/zone`, remove docs for `route53/hostedzone`. (issue 1837, issue 1837, commit 99e2e67)

4.44.2 BugFixes

- Fix for MWS iterator handling (commit 7e6f98d)
- Clarify documentation for `MetricAlarm` dimensions. (issue 1808, issue 1808, issue 1803, commit 4233fbf)
- Fixes for general connection behind proxy. (issue 1781, issue 1781, commit dc8bbea)
- Validate S3 method kwarg names to prevent misspelling. (issue 1810, issue 1810, issue 1782, commit 947a14a)
- Fix dependencies so they show up as optional in `CheeseShop` (issue 1617, commit 54da8b6)
- Route53 retry HTTP error 400s (issue 1618, commit 6e355b3)
- Fix typo in `IAMConnection` documentation (issue 1820, commit 3fc335d)
- Fix MWS `MemberLists` parsing. (issue 1815, issue 1815, commit 0f6f089)
- Fix typo in `SQS` documentation (issue 1830, commit 20532a6)
- Update auto scaling documentation. (issue 1824, issue 1824, issue 1823, commit 9a359ec)
- Fixing region endpoints for EMR (issue 1831, commit ed669f7)
- Raising an exception in `SQS` message `decode()` should not abort parsing. (issue 1835, issue 1835, issue 1833, commit 2a00c92)
- Replace correct VPC ACL association instead of just the first one. (issue 1844, issue 1844, issue 1843, commit c70b8d6)
- Prevent swallowing `CloudSearch` errors (issue 1846, issue 1842, commit c2f955b)

4.45 boto v2.15.0

date 2013/10/17

This release adds support for Amazon Elastic Transcoder audio transcoding, new regions for Amazon Simple Storage Service (S3), Amazon Glacier, and Amazon Redshift as well as new parameters in Amazon Simple Queue Service (SQS), Amazon Elastic Compute Cloud (EC2), and the `lss3` utility. Also included are documentation updates and fixes for S3, Amazon DynamoDB, Amazon Simple Workflow Service (SWF) and Amazon Marketplace Web Service (MWS).

4.45.1 Features

- Add SWF tutorial and code sample (issue 1769, commit 36524f5)
- Add ap-southeast-2 region to S3WebsiteEndpointTranslate (issue 1777, commit e7b0b39)
- Add support for `owner_acct_id` in SQS `get_queue` (issue 1786, commit c1ad303)
- Add ap-southeast-2 region to Glacier (commit c316266)
- Add ap-southeast-1 and ap-southeast-2 to Redshift (commit 3d67a03)
- Add SSH timeout option (issue 1755, commit d8e70ef, commit 653b82b)
- Add support for markers in `lss3` (issue 1783, commit 8ee4b1f)
- Add `block_device_mapping` to EC2 `create_image` (issue 1794, commit 86afe2e)
- Updated SWF tutorial (issue 1797, commit 3804b16)
- Support Elastic Transcoder audio transcoding (commit 03a5087)

4.45.2 Bugfixes

- Fix VPC module docs, ELB docs, some formatting (issue 1770, commit 75de377)
- Fix DynamoDB item `attrs` initialization (issue 1776, commit 8454a2b)
- Fix parsing of empty member lists for MWS (issue 1785, commit 7b46ca5)
- Fix link to release notes in docs (commit a6bf794)
- Do not validate bucket when copying a key (issue 1763, commit 5505113)
- Retry HTTP 502, 504 errors (issue 1798, commit c832e2d)

4.46 boto v2.14.0

date 2013/10/09

This release makes `s3put` region-aware, adds some missing features to EC2 and SNS, enables EPUB documentation output, and makes the HTTP(S) connection pooling port-aware, which in turn enables connecting to e.g. mock services running on `localhost`. It also includes support for the latest EC2 and OpsWorks features, as well as several important bugfixes for EC2, DynamoDB, MWS, and Python 2.5 support.

4.46.1 Features

- **Add support for a `--region` argument to `s3put` and auto-detect bucket** regions if possible (issue 1731, commit d9c28f6)
- **Add `delete_notification_configuration` for EC2 autoscaling** (issue 1717, commit ebb7ace)
- Add support for registering HVM instances (issue 1733, commit 2afc68e)
- **Add support for `ReplaceRouteTableAssociation` for EC2** (issue 1736, commit 4296835)
- Add `sms` as an option for SNS `subscribe` (issue 1744, commit 8ff08e5)
- Allow overriding `has_google_credentials` (issue 1752, commit 052cc91)
- Add EPUB output format for docs (issue 1759, commit def7c67)

- **Add handling of Connection: close HTTP headers in responses** (issue 1773, commit 1a38f32)
- **Make connection pooling port-aware** (issue 1764, issue 1737, commit b6c7330)
- **Add support for instance_type to modify_reserved_instances** (commit bf07eee)
- Add support for new OpsWorks features (commit f512898)

4.46.2 Bugfixes

- Remove erroneous `dry_run` parameter (issue 1729, commit 35a516e)
- **Fix task_list override in poll methods of SWF Deciders and Workers** (issue 1724, commit fa8d871)
- **Remove Content-Encoding header from metadata test** (issue 1735, commit c8b0130)
- **Fix the ability to override DynamoDBv2 host and port when creating connections** (issue 1734, commit 8d2b492)
- Fix `UnboundLocalError` (commit e0e6aeb)
- **self.rules is of type IPPermissionsList, remove takes no kwargs** (commit 3c56b3f)
- Nicer error messages for 403s (issue 1753, commit d3d9eab)
- Various documentation fixes (issue 1762, commit 76aef10)
- Various Python 2.5 fixes (commit 150aef6, commit 67ae9ff)
- **Prevent certificate tests from failing for non-govcloud accounts** (commit 2d3d9f6)
- Fix flaky resumable upload test (issue 1768, commit 6aa8ae2)
- **Force the Host HTTP header to fix an issue with older httplib** (commit 202c456)
- Blacklist S3 from forced Host HTTP header (commit 9193226)
- Fix `propagate_at_launch` spelling error (issue 1739, commit e78d88a)
- **Remove unused code that causes exceptions with bad response data** (issue 1771, commit bec5e70)
- Fix `detach_subnets` typo (issue 1760, commit 4424e1b)
- **Fix result list handling of GetMatchingProductForIdResponse for MWS** (issue 1751, commit 977b7dc)

4.47 boto v2.13.3

date 2013/09/16

This release fixes a packaging error with the previous version of boto. The version `v2.13.2` was provided instead of `2.13.2`, causing things like `pip` to incorrectly resolve the latest release.

That release was only available for several minutes & was removed from PyPI due to the way it would break installation for users.

4.48 boto v2.13.2

date 2013/09/16

This release is a bugfix-only release, correcting several problems in EC2 as well as S3, DynamoDB v2 & SWF.

Note: There was no v2.13.1 release made public. There was a packaging error that was discovered before it was published to PyPI.

We apologise for the fault in the releases. Those responsible have been sacked.

4.48.1 Bugfixes

- Fixed test fallout from the EC2 dry-run change. (commit 2159456)
- Added tests for more of SWF's `layer2`. (issue 1718, commit 35fb741, commit a84d401, commit 1cf1641, commit a36429c)
- Changed EC2 to allow `name` to be optional in calls to `copy_image`. (issue 1672, :sha:' 26285aa')
- Added `billingProducts` support to EC2 Image. (issue 1703, commit cccadaf, commit 3914e91)
- Fixed a place where `dry_run` was handled in EC2. (issue 1722, commit 0a52c82)
- Fixed `run_instances` with a block device mapping. (issue 1723, commit 974743f, commit 9049f05, commit d7eda6c)
- Fixed `s3put` to accept headers with a `=` in them. (issue 1700, commit 7958c70)
- Fixed a bug in DynamoDB v2 where scans with filters over large sets may not return all values. (issue 1713, commit 02893e1)
- Cloudsearch now uses SigV4. (commit b2bdf5)
- Several documentation improvements/fixes:
 - Added the “Apps Built On Boto” doc. (commit 3bd628c)

4.49 boto v2.13.0

date 2013/09/12

This release adds support for VPC within AWS Opsworks, added dry-run support & the ability to modify reserved instances in EC2 as well as several important bugfixes for EC2, SNS & DynamoDBv2.

4.49.1 Features

- Added support for VPC within Opsworks. (commit 56e1df3)
- Added support for `dry_run` within EC2. (commit dd7774c)
- Added support for `modify_reserved_instances` & `describe_reserved_instances_modifications` within EC2. (commit 7a08672)

4.49.2 Bugfixes

- Fixed EC2's `associate_public_ip` to work correctly. (commit 9db6101)
- Fixed a bug with `dynamodb_load` when working with sets. (issue 1664, commit ef2d28b)

- Changed SNS `publish` to use POST. (commit 9c11772)
- Fixed inability to create LaunchConfigurations when using Block Device Mappings. (issue 1709, issue 1710, commit 5fd728e)
- Fixed DynamoDBv2's `batch_write` to appropriately handle `UnprocessedItems`. (issue 1566, issue 1679, issue 1714, commit 2fc2369)
- Several documentation improvements/fixes:
 - Added Opsworks docs to the index. (commit 5d48763)
 - Added docs on the correct string values for `get_all_images`. (issue 1674, commit 1e4ed2e)
 - Removed a duplicate `boto.s3.prefix` entry from the docs. (issue 1707, commit b42d34c)
 - Added an API reference for `boto.swf.layer2`. (issue 1712, commit 9f7b15f)

4.50 boto v2.12.0

date 2013/09/04

This release adds support for Redis & replication groups to Elasticache as well as several bug fixes.

4.50.1 Features

- Added support for Redis & replication groups to Elasticache. (commit f744ff6)

4.50.2 Bugfixes

- Boto's User-Agent string has changed. Mostly additive to include more information. (commit edb038a)
- Headers that are part of S3's signing are now correctly coerced to the proper case. (issue 1687, commit 89eae8c)
- Altered S3 so that it's possible to track what portions of a multipart upload succeeded. (issue 1305, issue 1675, commit e9a2c59)
- Added `create_lb_policy` & `set_lb_policies_of_backend_server` to ELB. (issue 1695, commit 77a9458)
- Fixed pagination when listing vaults in Glacier. (issue 1699, commit 9afecca)
- Several documentation improvements/fixes:
 - Added some docs about what command-line utilities ship with boto. (commit 5d7d54d)

4.51 boto v2.11.0

date 2013/08/29

This release adds Public IP address support for VPCs created by EC2. It also makes the GovCloud region available for all services. Finally, this release also fixes a number of bugs.

4.51.1 Features

- Added Public IP address support within VPCs created by EC2. (commit [be132d1](#))
- All services can now easily use GovCloud. (issue [1651](#), commit [542a301](#), commit [3c56121](#), commit [9167d89](#))
- Added `db_subnet_group` to `RDSConnection.restore_dbinstance_from_point_in_time`. (issue [1640](#), commit [06592b9](#))
- Added `monthly_backups` to EC2's `trim_snapshots`. (issue [1688](#), commit [a2ad606](#), commit [2998c11](#), commit [e32d033](#))
- Added `get_all_reservations` & `get_only_instances` methods to EC2. (issue [1572](#), commit [ffc6cc0](#))

4.51.2 Bugfixes

- Fixed the parsing of CloudFormation's `LastUpdatedTime`. (issue [1667](#), :sha:' 70f363a')
- Fixed STS' `assume_role_with_web_identity` to work correctly. (issue [1671](#), commit [ed1f403](#), commit [ca794d5](#), commit [ed7e563](#), commit [859762d](#))
- Fixed how VPC security group filtering is done in EC2. (issue [1665](#), issue [1677](#), commit [be00956](#), commit [5e85dd1](#), commit [e63aae8](#))
- Fixed fetching more than 100 records with `ResourceRecordSet`. (issue [1647](#), issue [1648](#), issue [1680](#), commit [b64dd4f](#), commit [276df7e](#), commit [e57cab0](#), commit [e62a58b](#), commit [4c81bea](#), commit [a3c635b](#))
- Fixed how VPC Security Groups are referred to when working with RDS. (issue [1602](#), issue [1683](#), issue [1685](#), issue [1694](#), commit [012aa0c](#), commit [d5c6dfa](#), commit [7841230](#), commit [0a90627](#), commit [ed4fd8c](#), commit [61d394b](#), commit [ebe84c9](#), commit [a6b0f7e](#))
- Google Storage `Key` now uses transcoding-invariant headers where possible. (commit [d36eac3](#))
- Doing non-multipart uploads when using `s3put` no longer requires having the `ListBucket` permission. (issue [1642](#), issue [1693](#), commit [f35e914](#))
- Fixed the serialization of `attributes` in a variety of SNS methods. (issue [1686](#), commit [4afb3dd](#), commit [a58af54](#))
- Fixed SNS to be better behaved when constructing an mobile push notification. (issue [1692](#), commit [62fdf34](#))
- Moved SWF to SigV4. (commit [ef7d255](#))
- Several documentation improvements/fixes:
 - Updated the DynamoDB v2 docs to correct how the connection is built. (issue [1662](#), commit [047962d](#))
 - Fixed a typo in the DynamoDB v2 docstring for `Table.create`. (commit [be00956](#))
 - Fixed a typo in the DynamoDB v2 docstring for `Table` for custom connections. (issue [1681](#), commit [6a53020](#))
 - Fixed incorrect parameter names for `DBParameterGroup` in RDS. (issue [1682](#), commit [0d46aed](#))
 - Fixed a typo in the SQS tutorial. (issue [1684](#), commit [38b7889](#))

4.52 boto v2.10.0

date 2013/08/13

This release adds Mobile Push Notification support to Amazon Simple Notification Service, better reporting for Amazon Redshift, SigV4 authorization for Amazon Elastic MapReduce & lots of bugfixes.

4.52.1 Features

- Added support for Mobile Push Notifications to SNS. This enables you to send push notifications to mobile devices (such as iOS or Android) using SNS. (commit [ccba574](#))
- Added support for better reporting within Redshift. (commit [9d55dd3](#))
- Switched Elastic MapReduce to use SigV4 for authorization. (commit [b80aa48](#))

4.52.2 Bugfixes

- Added the `MinAdjustmentType` parameter to EC2 Autoscaling. (issue [1562](#), issue [1619](#), commit [1760284](#), commit [2a11fd9](#), commit [2d14006](#) & commit [b7f1ae1](#))
- Fixed how DynamoDB tracks changes to data in `Item` objects, fixing failures with modified sets not being sent. (issue [1565](#), commit [b111fcf](#) & commit [812f9a6](#))
- Updated the CA certificates Boto ships with. (issue [1578](#), commit [4dfadc8](#))
- Fixed how CloudSearch's `Layer2` object gets initialized. (issue [1629](#), issue [1630](#), commit [40b3652](#) & commit [f797ff9](#))
- Fixed the `-w` flag in `s3put`. (issue [1637](#), commit [0865004](#) & commit [3fe70ca](#))
- Added the `ap-southeast-2` endpoint for DynamoDB. (issue [1621](#), commit [501b637](#))
- Fixed test suite to run faster. (commit [243a67e](#))
- Fixed how non-JSON responses are caught from CloudSearch. (issue [1633](#), issue [1645](#), commit [d5a5c01](#), commit [954a50c](#), commit [915d8ff](#) & commit [4407fcb](#))
- Fixed how `DeviceIndex` is parsed from EC2. (issue [1632](#), issue [1646](#), commit [ff15e1f](#), commit [8337a0b](#) & commit [27c9b04](#))
- Fixed EC2's `connect_to_region` to respect the `region` parameter. (issue [1616](#), issue [1654](#), commit [9c37256](#), commit [5950d12](#) & commit [b7eebe8](#))
- Added `modify_network_interface_attribute` to EC2 connections. (issue [1613](#), issue [1656](#), commit [e00b601](#), commit [5b62f27](#), commit [126f6e9](#), commit [bbfed1f](#) & commit [0c61293](#))
- Added support for `param_group` within RDS. (issue [1639](#), commit [c47baf0](#))
- Added support for using `Item.partial_save` to create new records within DynamoDBv2. (issue [1660](#), issue [1521](#), commit [bfa469f](#) & commit [58a13d7](#))
- Several documentation improvements/fixes:
 - Updated guideline on how core should merge PRs. (commit [80a419c](#))
 - Fixed a typo in a CloudFront docstring. (issue [1657](#), commit [1aa0621](#))

4.53 boto v2.9.9

date 2013/07/24

This release updates Opworks to add AMI & Chef 11 support, DBSubnetGroup support in RDS & many other bugfixes.

4.53.1 Features

- Added AMI, configuration manager & Chef 11 support to Opsworks. (commit 55725fc).
- Added `in` support for SQS messages. (issue 1593, commit e5fe1ed)
- Added support for the `ap-southeast-2` region in Elasticache. (issue 1607, commit 9986b61)
- Added support for block device mappings in ELB. (issue 1343, issue 753, issue 1357, commit 974a23a)
- Added support for `DBSubnetGroup` in RDS. (issue 1500, commit 01eef87, commit 45c60a0, commit c4c859e)

4.53.2 Bugfixes

- Fixed the canonicalization of paths on Windows. (issue 1609, commit a1fa98c)
- Fixed how `BotoServerError` uses `message`. (issue 1353, commit b944f4b)
- Fixed `DisableRollback` always being `True` in a `CloudFormation Stack`. (issue 1379, commit 32b3150)
- Changed EMR instance groups to no longer require a string price (can now be a `Decimal`). (issue 1396, commit dfc39ff)
- Altered `Distribution._sign_string` to accept any file-like object as well within `CloudFront`. (issue 1349, commit 8df6c14)
- Fixed the `detach_lb_from_subnets` call within ELB. (issue 1417, issue 1418 commit 4a397bd, commit c11d72b, commit 9e595b5, commit 634469d, commit 586dd54)
- Altered boto to obey `no_proxy` environment variables. (issue 1600, issue 1603, commit aaf5a9)
- Fixed ELB connections to use HTTPS by default. (issue 1587, commit fe158c4)
- Updated S3 to be Python 2.5 compatible again. (issue 1598, commit 066009f)
- All calls within SES will now return *all* DKIMTokens, instead of just one. (issue 1550, issue 1610, commit 1a079da, commit 1e82f85, commit 5c8b6b8)
- Fixed the logging parameter within `DistributionConfig` in `CloudFront` to respect whatever is provided to the constructor. (issue 1457, commit e76180d)
- Fixed `CloudSearch` to no longer raise an error if a non-JSON response is received. (issue 1555, issue 1614, commit 5e2c292, commit 6510e1f)

4.54 boto v2.9.8

date 2013/07/18

This release adds new methods in AWS Security Token Service (STS), AWS CloudFormation, updates AWS Relational Database Service (RDS) & Google Storage. It also has several bugfixes & documentation improvements.

4.54.1 Features

- Added support for the `DecodeAuthorizationMessage` in STS (commit 1ada5ac).
- Added support for creating/deleting/describing `OptionGroup` in RDS. (commit d629228 & commit d059a3b)
- Added `CancelUpdateStack` to `CloudFormation`. (issue 1476, commit 5bae130)
- Added support for getting/setting lifecycle configurations on GS buckets. (issue 1604, commit 652fc81)

4.54.2 Bugfixes

- Added region support to `bin/elbadmin`. (issue 1586, commit 2ffbc60)
- Changed the mock storage to use case-insensitive headers. (issue 1594, commit 71849cb)
- Added `complex_listeners` to ELB. (issue 1048, commit b782ce2)
- Added tests for Route53's `ResourceRecordSets`. (commit fad5bde)
- Several documentation improvements/fixes:
 - Updated CloudFront docs. (issue 1546, commit a811197)
 - Updated the URL explaining the use of base64 in SQS messages. (issue 1596, commit 00de3a2)

4.55 boto v2.9.7

date 2013/07/08

This release is primarily a bugfix release, but also includes support for Elastic Transcoder updates (variable bit rate, max frame rate & watermark features).

4.55.1 Features

- Added support for selecting specific attributes in DynamoDB v2. (issue 1567, commit d9e5c2)
- Added support for variable bit rate, max frame rate & watermark features in Elastic Transcoder. (commit 3791c9)

4.55.2 Bugfixes

- Altered RDS to now use SigV4. (commit be1633)
- Removed parsing check in `StorageUri`. (commit 21bc8f)
- More information returned about GS key generation. (issue 1571, commit 6d5e3a)
- Upload handling headers now case-insensitive. (issue 1575, commit 60383d)
- Several CloudFormation timestamp updates. (issue 1582, issue 1583, issue 1588, commit 0a23d34, commit 6d4209)
- Corrected a bug in how limits are handled in DynamoDB v2. (issue 1590, commit 710a62)
- Several documentation improvements/fixes:
 - Typo in `boto.connection` fixed. (issue 1569, commit cf39fd)
 - All previous release notes added to the docs. (commit 165596)
 - Corrected error in `get_all_tags` docs. (commit 4bca5d)
 - Corrected a typo in the S3 tutorial. (commit f0cef8)
 - Corrected several import errors in the DDBv2 tutorial. (commit 5401a3)
 - Fixed an error in the `get_key_pair` docstring. (issue 1590, commit a9cb8d)

4.56 boto v2.9.6

date 2013/06/18

This release adds large payload support to Amazon SNS/SQS (from 32k to 256k bodies), several minor API additions, new regions for Redshift/Cloudsearch & a host of bugfixes.

4.56.1 Features

- Added large body support to SNS/SQS. There's nothing to change in your application code, but you can now send payloads of up to 256k in size. (commit [b64947](#))
- Added `Vault.retrieve_inventory_job` to Glacier. (issue [1532](#), commit [33de29](#))
- Added `Item.get(...)` support to DynamoDB v2. (commit [938cb6](#))
- Added the `ap-northeast-1` region to Redshift. (commit [d3eb61](#))
- Added all the current regions to Cloudsearch. (issue [1465](#), commit [22b3b7](#))

4.56.2 Bugfixes

- Fixed a bug where date metadata couldn't be set on an S3 key. (issue [1519](#), commit [1efde8](#))
- Fixed Python 2.5/Jython support in `NetworkInterfaceCollection`. (issue [1518](#), commit [0d6af2](#))
- Fixed a XML parsing error with `InstanceStatusSet`. (issue [1493](#), commit [55d4f6](#))
- Added a test case to try to demonstrate issue [443](#). (commit [084dd5](#))
- Exposed the current tree-hash & upload size on Glacier's `Writer`. (issue [1520](#), commit [ade462](#))
- Updated EC2 Autoscale to incorporate new cron-like parameters. (issue [1433](#), commit [266e25](#), commit [871588](#) & commit [473e42](#))
- Fixed `AttributeError` being thrown from `LoadBalancerZones`. (issue [1524](#), commit [215ffa](#))
- Fixed a bug with empty facets in Cloudsearch. (issue [1366](#), commit [7a108e](#))
- Fixed an S3 timeout/retry bug where HTTP 400s weren't being honored. (issue [1528](#), commit [efd9af](#) & commit [16ae74](#))
- Fixed `get_path` when `suppress_consec_slashes=False`. (issue [1522](#), commit [c5dff6](#))
- Factored out how some of S3's `query_args` are constructed. (commit [9f73de](#))
- Added the `generation` query param to `gs.Key.open_read`. (commit [cb4427](#))
- Fixed a bug with the canonicalization of URLs with trailing slashes in the SigV4 signer. (issue [1541](#), commit [dec541](#), commit [3f2b33](#))
- Several documentation improvements/fixes:
 - Updated the release notes slightly. (commit [7b6079](#))
 - Corrected the `num_cb` param on `set_contents_from_filename`. (issue [1523](#), commit [44be69](#))
 - Fixed some example code in the DDB migration guide. (issue [1525](#), commit [6210ca](#))
 - Fixed a typo in one of the DynamoDB v2 examples. (issue [1551](#), commit [b0df3e](#))

4.57 boto v2.9.5

date 2013/05/28

This release adds support for [web identity federation](#) within the Secure Token Service (STS) & fixes several bugs.

4.57.1 Features

- Added support for web identity federation - You can now delegate token access via either an Oauth 2.0 or OpenID provider. ([commit 9bd0a3](#))

4.57.2 Bugfixes

- Altered the S3 key buffer to be a configurable value. ([issue 1506](#), [commit 8e3e36](#))
- Added Sphinx extension for better release notes. ([issue 1511](#), [commit e2e32d](#) & [commit 3d998b](#))
- Fixed a bug where DynamoDB v2 would only ever connect to the default endpoint. ([issue 1508](#), [commit 139912](#))
- Fixed a iteration/empty results bug & a `between` bug in DynamoDB v2. ([issue 1512](#), [commit d109b6](#))
- Fixed an issue with `EbsOptimized` in EC2 Autoscale. ([issue 1513](#), [commit 424c41](#))
- Fixed a missing instance variable bug in DynamoDB v2. ([issue 1516](#), [commit 6fa8bf](#))

4.58 boto v2.9.4

date 2013/05/20

This release adds updated Elastic Transcoder support & fixes several bugs from recent releases & API updates.

4.58.1 Features

- Updated Elastic Transcoder support - It now supports HLS, WebM, MPEG2-TS & a host of [other features](#). ([commit 89196a](#))

4.58.2 Bugfixes

- Fixed a bug in the canonicalization of URLs on Windows. ([commit 09ef8c](#))
- Fixed glacier part size bug ([issue 1478](#), [commit 9e04171](#))
- Fixed a bug in the bucket regex for S3 involving capital letters. ([commit 950031](#))
- Fixed a bug where timestamps from Cloudformation would fail to be parsed. ([commit b40542](#))
- Several documentation improvements/fixes:
 - Added autodocs for many of the EC2 apis. ([commit 79f939](#))

4.59 boto v2.9.3

date 2013/05/15

This release adds ELB support to Opsworks, optimized EBS support in EC2 AutoScale, Parallel Scan support to DynamoDB v2, a higher-level interface to DynamoDB v2 and API updates to DataPipeline.

4.59.1 Features

- ELB support in Opsworks - You can now attach & describe the Elastic Load Balancers within the Opsworks client. (commit [ecda87](#))
- Optimized EBS support in EC2 AutoScale - You can now specify whether an AutoScale instance should be optimized for EBS I/O. (commit [f8acaa](#))
- Parallel Scan support in DynamoDB v2 - If you have extra read capacity & a large amount of data, you can scan over the records in parallel by telling DynamoDB to split the table into segments, then spinning up threads/processes to each run over their own segment. (commit [db7f7b](#) & commit [7ed73c](#))
- Higher-level interface to DynamoDB v2 - A more convenient API for using DynamoDB v2. The *DynamoDB v2 Tutorial* has more information on how to use the new API. (commit [0f7c8b](#))

4.59.2 Backward-Incompatible Changes

- API Update for DataPipeline - The `error_code` (integer) argument to `set_task_status` changed to `error_id` (string). Many documentation updates were also added. (commit [a78572](#))

4.59.3 Bugfixes

- Bumped the AWS Support API version. (commit [0323f4](#))
- Fixed the S3 ResumableDownloadHandler so that it no longer tries to use a hashing algorithm when used outside of GCS. (commit [29b046](#))
- Fixed a bug where Sig V4 URIs were improperly canonicalized. (commit [5269d8](#))
- Fixed a bug where Sig V4 ports were not included. (commit [cfaba3](#))
- Fixed a bug in CloudWatch's `build_put_params` that would overwrite existing/necessary variables. (commit [550e00](#))
- Several documentation improvements/fixes:
 - Added docs for RDS `modify/modify_dbinstance`. (commit [777d73](#))
 - Fixed a typo in the `README.rst`. (commit [181e0f](#))
 - Documentation fallout from the previous release. (commit [14a111](#))
 - Fixed a typo in the EC2 `Image.run` docs. (commit [5edd6a](#))
 - Added/improved docs for EC2 `Image.run`. (commit [773ce5](#))
 - Added a CONTRIBUTING doc. (commit [cecb8](#))
 - Fixed S3 `create_bucket` docs to specify “European Union”. (commit [ddddfd](#))

4.60 boto v2.9.2

date 2013/04/30

A hotfix release that adds the `boto.support` into `setup.py`.

4.60.1 Features

- None.

4.60.2 Bugfixes

- Fixed the missing `boto.support` in `setup.py`. ([commit 9ac196](#))

4.61 boto v2.9.1

date 2013/04/30

Primarily a bugfix release, this release also includes support for the new AWS Support API.

4.61.1 Features

- AWS Support API - A client was added to support the new AWS Support API. It gives programmatic access to Support cases opened with AWS. A short example might look like:

```
>>> from boto.support.layer1 import SupportConnection
>>> conn = SupportConnection()
>>> new_case = conn.create_case(
...     subject='Description of the issue',
...     service_code='amazon-cloudsearch',
...     category_code='performance',
...     communication_body="We're seeing some latency from one of our...",
...     severity_code='low'
... )
>>> new_case['caseId']
u'case-...'
```

The *Support Tutorial* has more information on how to use the new API. ([commit 8c0451](#))

4.61.2 Bugfixes

- The reintroduction of `ResumableUploadHandler.get_upload_id` that was accidentally removed in a previous commit. ([commit 758322](#))
- Added `OrdinaryCallingFormat` to support Google Storage's certificate verification. ([commit 4ca83b](#))
- Added the `eu-west-1` region for Redshift. ([commit e98b95](#))
- Added support for overriding the port any connection in `boto` uses. ([commit 08e893](#))
- Added retry/checksumming support to the DynamoDB v2 client. ([commit 969ae2](#))
- Several documentation improvements/fixes:

- Incorrect docs on EC2's `import_key_pair`. (commit 6ada7d)
- Clearer docs on the DynamoDB `count` parameter. (commit dfa456)
- Fixed a typo in the `autoscale_tut`. (commit 6df1ae)

4.62 boto v2.9.0

The 2.9.0 release of boto is now available on PyPI.

You can get a comprehensive list of all commits made between the 2.8.0 release and the 2.9.0 release at <https://github.com/boto/boto/compare/2.8.0...2.9.0>.

This release includes:

- Support for Amazon Redshift
- Support for Amazon DynamoDB's new API
- Support for AWS Opsworks
- Add `copy_image` to EC2 (AMI copy)
- Add `describe_account_attributes` and `describe_vpc_attribute`, and `modify_vpc_attribute` operations to EC2.

There were 240 commits made by 34 different authors:

- g2harris
- Michael Barrett
- Pascal Hakim
- James Saryerwinnie
- Mitch Garnaat
- ChangMin Jeon
- Mike Schwartz
- Jeremy Katz
- Alex Schoof
- reinhillmann
- Travis Hobrla
- Zach Wilt
- Daniel Lindsley
- ksacry
- Michael Wirth
- Eric Smalling
- pingwin
- Chris Moyer
- Olivier Hervieu
- Iuri de Silvio
- Joe Sondow

- Max Noel
- Nate
- Chris Moyer
- Lars Otten
- Nathan Grigg
- Rein Hillmann
- Øyvind Saltvik
- Rayson HO
- Martin Matusiak
- Royce Remer
- Jeff Terrace
- Yaniv Ovadia
- Eduardo S. Klein

4.63 boto v2.8.0

The 2.8.0 release of boto is now available on [PyPI](#).

You can get a comprehensive list of all commits made between the 2.7.0 release and the 2.8.0 release at <https://github.com/boto/boto/compare/2.7.0...2.8.0>.

This release includes:

- Added support for Amazon ElastiCache
- Added support for Amazon Elastic Transcoding Service

As well as numerous bug fixes and improvements.

4.63.1 Commits

There were 115 commits in this release from 21 different authors. The authors are listed below, in alphabetical order:

- conorbranagan
- dkavanagh
- gaige
- garnaat
- halfaleague
- jamesls
- jjhooper
- jordansissel
- jterrace
- Kodiologist

- kopertop
- mfschwartz
- nathan11g
- pasc
- phobologic
- schworer
- seandst
- SirAlvarex
- Yaniv Ovadia
- yig
- yovadia12

4.64 boto v2.7.0

The 2.7.0 release of boto is now available on PyPI.

You can get a comprehensive list of all commits made between the 2.6.0 release and the 2.7.0 release at <https://github.com/boto/boto/compare/2.6.0...2.7.0>.

This release includes:

- Added support for AWS Data Pipeline - [commit 999902](#)
- Integrated Slick53 into Route53 module - [issue 1186](#)
- Add ability to use Decimal for DynamoDB numeric types - [issue 1183](#)
- Query/Scan Count/ScannedCount support and TableGenerator improvements - [issue 1181](#)
- Added support for keyring in config files - [issue 1157](#)
- Add concurrent downloader to glacier - [issue 1106](#)
- Add support for tagged RDS DBInstances - [issue 1050](#)
- Updating RDS API Version to 2012-09-17 - [issue 1033](#)
- Added support for provisioned IOPS for RDS - [issue 1028](#)
- Add ability to set SQS Notifications in Mechanical Turk - [issue 1018](#)

4.64.1 Commits

There were 447 commits in this release from 60 different authors. The authors are listed below, in alphabetical order:

- acrefoot
- Alex Schoof
- Andy Davidoff
- anoopj
- Benoit Dubertret

- bobveznat
- dahlia
- dangra
- disruptek
- dmccritchie
- emtrane
- focus
- fsouza
- g2harris
- garnaat
- georgegoh
- georgesequeira
- GitsMcGee
- glance-
- gtaylor
- hashbackup
- hinnerk
- hoov
- isaacbowen
- jamesls
- JerryKwan
- jimfulton
- jimbrowne
- jorourke
- jterrace
- jtriley
- katzj
- kennu
- kevinburke
- khagler
- Kodiologist
- kopertop
- kotnik
- Leftium
- lpetc
- marknca

- matthewandrews
- mfschwartz
- mikek
- mkmt
- mleonhard
- mraposa
- oozie
- phunter
- potix2
- Rafael Cunha de Almeida
- reinhillmann
- reversefold
- Robie Basak
- seandst
- siroken3
- staer
- tpodowd
- vladimir-sol
- yovadia12

4.65 boto v2.6.0

The 2.6.0 release of boto is now available on [PyPI](#).

You can get a comprehensive list of all commits made between the 2.5.2 release and the 2.6.0 release at <https://github.com/boto/boto/compare/2.5.2...2.6.0>.

This release includes:

- Support for Amazon Glacier
- Support for AWS Elastic Beanstalk
- CORS support for Amazon S3
- Support for Reserved Instances Resale in Amazon EC2
- Support for IAM Roles

4.65.1 SSL Certificate Verification

In addition, this release of boto changes the default behavior with respect to SSL certificate verification. Our friends at Google contributed code to boto well over a year ago that implemented SSL certificate verification. At the time, we felt the most prudent course of action was to make this feature an opt-in but we always felt that at some time in the future we would enable cert verification as the default behavior. Well, that time is now!

However, in implementing this change, we came across a bug in Python for all versions prior to 2.7.3 (see <http://bugs.python.org/issue13034> for details). The net result of this bug is that Python is able to check only the commonName in the SSL cert for verification purposes. Any subjectAltNames are ignored in large SSL keys. So, in addition to enabling verification as the default behavior we also changed some of the service endpoints in boto to match the commonName in the SSL certificate.

If you want to disable verification for any reason (not advised, btw) you can still do so by editing your boto config file (see <https://gist.github.com/3762068>) or you can override it by passing `validate_certs=False` to the Connection class constructor or the `connect_*` function.

4.65.2 Commits

There were 440 commits in this release from 53 different authors. The authors are listed below, in alphabetical order:

- acorley
- acrefoot
- aedeph
- allardhoeve
- almost
- awatts
- buzztroll
- cadams
- cbednarski
- cosmin
- dangra
- darjus-amzn
- disruptek
- djw
- garnaat
- gertjanol
- gimbel0893
- gochist
- graphaelli
- gtaylor
- gz
- hardys
- jamesls
- jijoiv
- jimbrowne
- jtlebigot
- jtriley

- kopertop
- kotnik
- marknca
- mark_nunnikhoven
- mfschwartz
- moliware
- NeilW
- nkvoll
- nsitarz
- ohe
- pasieronon
- patricklucas
- pfig
- rajivnavada
- reversefold
- robie
- scott
- shawnp
- smoser
- sopel
- staer
- tedder
- yamatt
- Yossi
- yovadia12
- zachhuff386

4.66 boto v2.5.2

Release 2.5.2 is a bugfix release. It fixes the following critical issues: * [issue 830](#)

This issue only affects you if you are using DynamoDB on an EC2 instance with IAM Roles.

4.67 boto v2.5.1

Release 2.5.1 is a bugfix release. It fixes the following critical issues: * [issue 819](#)

4.68 boto v2.5.0

The 2.5.0 release of boto is now available on PyPI.

You can get a comprehensive list of all commits made between the 2.4.1 release and the 2.5.0 release at <https://github.com/boto/boto/compare/2.4.1...2.5.0>.

This release includes:

- Support for IAM Roles for EC2 Instances
- Added support for Capabilities in CloudFormation
- Spot instances in autoscaling groups
- Internal ELB's
- Added tenancy option to run_instances

There were 77 commits in this release from 18 different authors. The authors are listed below, in no particular order:

- jimbrowne
- cosmin
- gtaylor
- garnaat
- brianjaystanley
- jamesls
- trevorsummerssmith
- Bryan Donlan
- davidmarble
- jtriley
- rdodev
- toby
- tpodowd
- srs81
- mfschwartz
- rdegges
- gholms

4.69 boto v2.4.0

The 2.4.0 release of boto is now available on PyPI.

You can get a comprehensive list of all commits made between the 2.3.0 release and the 2.4.0 release at <https://github.com/boto/boto/compare/2.3.0...2.4.0>.

This release includes:

- Initial support for Amazon Cloudsearch Service.

- Support for Amazon's Marketplace Web Service.
- Latency-based routing for Route53
- Support for new domain verification features of SES.
- A full rewrite of the FPS module.
- Support for BatchWriteItem in DynamoDB.
- Additional EMR steps for installing and running Pig scripts.
- Support for additional batch operations in SQS.
- Better support for VPC group-ids.
- Many, many bugfixes from the community. Thanks for the reports and pull requests!

There were 175 commits in this release from 32 different authors. The authors are listed below, in no particular order:

- estebistec
- tpodowd
- Max Noel
- garnaat
- mfschwartz
- jtriley
- akoumjian
- jreese
- mulka
- Nuutti Kotivuori
- mboersma
- ryansb
- dampier
- crschmidt
- nithint
- sievlev
- eckamm
- imlucas
- disruptek
- trevorsummerssmith
- tmorgan
- evanworley
- iandanforth
- oozie
- aedeph
- alexanderdean

- abrinsmead
- dlecocq
- bsimpson63
- jamesls
- cosmin
- gtaylor

4.70 boto v2.3.0

The 2.3.0 release of boto is now available on PyPI.

You can view a list of issues that have been closed in this release at <https://github.com/boto/boto/issues?milestone=6&state=closed>.

You can get a comprehensive list of all commits made between the 2.2.2 release and the 2.3.0 release at <https://github.com/boto/boto/compare/2.2.2...2.3.0>.

This release includes initial support for Amazon Simple Workflow Service.

The API version of the FPS module was updated to 2010-08-28.

This release also includes many bug fixes and improvements in the Amazon DynamoDB module. One change of particular note is the behavior of the `new_item` method of the `Table` object. See <http://readthedocs.org/docs/boto/en/2.3.0/ref/dynamodb.html#module-boto.dynamodb.table> for more details.

There were 109 commits in this release from 21 different authors. The authors are listed below, in no particular order:

- theju
- garnaat
- rdodev
- mfschwartz
- kopertop
- tpodowd
- gtaylor
- kachok
- croach
- tmorgan
- Erick Fejta
- dherbst
- marccohen
- Arif Amirani
- yuzeh
- Roguelazer
- awblocker
- blinsay

- Peter Broadwell
- tierney
- georgekola

4.71 boto v2.2.2

The 2.2.2 release of boto is now available on PyPI.

You can view a list of issues that have been closed in this release at <https://github.com/boto/boto/issues?milestone=8&state=closed>.

You can get a comprehensive list of all commits made between the 2.2.1 release and the 2.2.2 release at <https://github.com/boto/boto/compare/2.2.1...2.2.2>.

This is a bugfix release.

There were 71 commits in this release from 11 different authors. The authors are listed below, in no particular order:

- aficionado
- jimbrowne
- rdodev
- milancermak
- garnaat
- kopertop
- samuraisam
- tpodowd
- psa
- mfschwartz
- gtaylor

4.72 boto v2.2.1

The 2.2.1 release fixes a packaging problem that was causing problems when installing via pip.

4.73 boto v2.2.0

The 2.2.0 release of boto is now available on PyPI.

You can view a list of issues that have been closed in this release at <https://github.com/boto/boto/issues?milestone=5&state=closed>.

You can get a comprehensive list of all commits made between the 2.0 release and the 2.1.0 release at <https://github.com/boto/boto/compare/fa0d6a1e49c8468abbe2c99cdc9f5fd8fd19f8f8...26c8eb108873bf8ce1b9d96d642eea2beef78c77>

Some highlights of this release:

- Support for Amazon DynamoDB service.

- Support for S3 Object Lifecycle (Expiration).
- Allow anonymous request for S3.
- Support for creating Load Balancers in VPC.
- Support for multi-dimension metrics in CloudWatch.
- Support for Elastic Network Interfaces in EC2.
- Support for Amazon S3 Multi-Delete capability.
- Support for new AMIversion and overriding of parameters in EMR.
- Support for SendMessageBatch request in SQS.
- Support for DescribeInstanceStatus request in EC2.
- Many, many improvements and additions to API documentation and Tutorials. Special thanks to Greg Taylor for all of the Sphinx cleanups and new docs.

There were 336 commits in this release from 40 different authors. The authors are listed below, in no particular order:

- Garrett Holmstrom
- mLewisLogic
- Warren Turkal
- Nathan Binkert
- Scott Moser
- Jeremy Edberg
- najeira
- Marc Cohen
- Jim Browne
- Mitch Garnaat
- David Ormsbee
- Blake Maltby
- Thomas O'Dowd
- Victor Trac
- David Marin
- Greg Taylor
- rdodev
- Jonathan Sabo
- rdoc
- Mike Schwartz
- l33twolf
- Keith Fitzgerald
- Oleksandr Gituliar
- Jason Allum

- Ilya Volodarsky
- Rajesh
- Felipe Reyes
- Andy Grimm
- Seth Davis
- Dave King
- andy
- Chris Moyer
- ruben
- Spike Gronim
- Daniel Norberg
- Justin Riley
- Milan Cermak timtebeek
- unknown
- Yotam Gingold
- Brian Oldfield

We processed 21 pull requests for this release from 40 different contributors. Here are the github user id's for all of the pull request authors:

- milancermak
- jsabo
- gituliar
- rdodev
- marccohen
- tpodowd
- trun
- jallum
- binkert
- ormsbee
- timtebeek

4.74 boto v2.1.1

The 2.1.1 release fixes one serious issue with the RDS module.

<https://github.com/boto/boto/issues/382>

4.75 boto v2.1.0

The 2.1.0 release of boto is now available on [PyPI](#) and [Google Code](#).

You can view a list of issues that have been closed in this release at <https://github.com/boto/boto/issues?milestone=4&state=closed>)

You can get a comprehensive list of all commits made between the 2.0 release and the 2.1.0 release at <https://github.com/boto/boto/compare/033457f30d...a0a1fd54ef>.

Some highlights of this release:

- Server-side encryption now supported in S3.
- Better support for VPC in EC2.
- Support for combiner in StreamingStep for EMR.
- Support for CloudFormations.
- Support for streaming uploads to Google Storage.
- Support for generating signed URL's in CloudFront.
- MTurk connection now uses HTTPS by default, like all other Connection objects.
- You can now PUT multiple data points to CloudWatch in one call.
- CloudWatch Dimension object now correctly supports multiple values for same dimension name.
- Lots of documentation fixes/additions

There were 235 commits in this release from 35 different authors. The authors are listed below, in no particular order:

- Erick Fejta
- Joel Barciauskas
- Matthew Tai
- Hyunjung Park
- Mitch Garnaat
- Victor Trac
- Andy Grimm
- ZerothAngel
- Dan Lecocq
- jmallen
- Greg Taylor
- Brian Grossman
- Marc Brinkmann
- Hunter Blanks
- Steve Johnson
- Keith Fitzgerald
- Kamil Klimkiewicz
- Eddie Hebert

- garnaat
- Samuel Lucidi
- Kazuhiro Ogura
- David Arthur
- Michael Budde
- Vineeth Pillai
- Trevor Pounds
- Mike Schwartz
- Ryan Brown
- Mark
- Chetan Sarva
- Dan Callahan
- INADA Naoki
- Mitchell Hashimoto
- Chris Moyer
- Riobard
- Ted Romer
- Justin Riley
- Brian Beach
- Simon Ratner

We processed 60 pull requests for this release from 40 different contributors. Here are the github user id's for all of the pull request authors:

- jtriley
- mbr
- jbarciauskas
- hyunjung
- bugi
- ryansb
- gtaylor
- ehazlett
- secretmike
- riobard
- simonratner
- irskep
- sanbornm
- methane

- jumping
- mansam
- miGlanz
- dlecocq
- fdr
- mitchellh
- ehebert
- memory
- hblanks
- mbudde
- ZerothAngel
- goura
- natedub
- tpounds
- bwbeach
- mumrah
- chetan
- jmallen
- a13m
- mtai
- fejta
- jibs
- callahad
- vineethrp
- JDrosdeck
- gholms

If you are trying to reconcile that data (i.e. 35 different authors and 40 users with pull requests), well so am I. I'm just reporting on the data that I get from the Github api 8^)

4.76 Release Notes for boto 2.0

4.76.1 Highlights

There have been many, many changes since the 2.0b4 release. This overview highlights some of those changes.

- Fix connection pooling bug: don't close before reading.
- Added AddInstanceGroup and ModifyInstanceGroup to boto.emr
- Merge pull request #246 from chetan/multipart_s3put

- AddInstanceGroupsResponse class to boto.emr.emrobject.
- Removed extra print statement
- Merge pull request #244 from ryansb/master
- Added add_instance_groups function to boto.emr.connection. Built some helper methods for it, and added AddInstanceGroupsResponse class to boto.emr.emrobject.
- Added a new class, InstanceGroup, with just a __init__ and __repr__.
- Adding support for GetLoginProfile request to IAM. Removing commented lines in connection.py. Fixes GoogleCode issue 532.
- Fixed issue #195
- Added correct sax reader for boto.emr.emrobject.BootstrapAction
- Fixed a typo bug in ConsoleOutput sax parsing and some PEP8 cleanup in connection.py.
- Added initial support for generating a registration url for the aws marketplace
- Fix add_record and del_record to support multiple values, like change_record does
- Add support to accept SecurityGroupId as a parameter for ec2 run instances. This is required to create EC2 instances under VPC security groups
- Added support for aliases to the add_change method of ResourceRecordSets.
- Resign each request in a retry situation. Some services are starting to incorporate replay detection algorithms and the boto approach of simply re-trying the original request triggers them. Also a small bug fix to roboto and added a delay in the ec2 test to wait for consistency.
- Fixed a problem with InstanceMonitoring parameter of LaunchConfigurations for autoscale module.
- Route 53 Alias Resource Record Sets
- Fixed App Engine support
- Fixed incorrect host on App Engine
- Fixed issue 199 on github.
- First pass at put_metric_data
- Changed boto.s3.Bucket.set_acl_xml() to ISO-8859-1 encode the Unicode ACL text before sending over HTTP connection.
- Added GetQualificationScore for mturk.
- Added UpdateQualificationScore for mturk
- import_key_pair base64 fix
- Fixes for ses send_email method better handling of exceptions
- Add optional support for SSL server certificate validation.
- Specify a reasonable socket timeout for httplib
- Support for ap-northeast-1 region
- Close issue #153
- Close issue #154
- we must POST autoscale user-data, not GET. otherwise a HTTP 505 error is returned from AWS. see: http://groups.google.com/group/boto-dev/browse_thread/thread/d5eb79c97ea8eecf?pli=1

- autoscale userdata needs to be base64 encoded.
- Use the unversioned streaming jar symlink provided by EMR
- Updated lss3 to allow for prefix based listing (more like actual ls)
- Deal with the groupSet element that appears in the instanceSet element in the DescribeInstances response.
- Add a change_record command to bin/route53
- Incorporating a patch from AWS to allow security groups to be tagged.
- Fixed an issue with extra headers in generated URLs. Fixes <http://code.google.com/p/boto/issues/detail?id=499>
- Incorporating a patch to handle obscure bug in apache/fastcgi. See <http://goo.gl/0Tdax>.
- Reorganizing the existing test code. Part of a long-term project to completely revamp and improve boto tests.
- Fixed an invalid parameter bug (ECS) #102
- Adding initial cut at s3 website support.

4.76.2 Stats

- 465 commits since boto 2.0b4
- 70 authors
- 111 Pull requests from 64 different authors

4.76.3 Contributors (in order of last commits)

- Mitch Garnaat
- Chris Moyer
- Garrett Holmstrom
- Justin Riley
- Steve Johnson
- Sean Talts
- Brian Beach
- Ryan Brown
- Chetan Sarva
- spenczar
- Jonathan Drosdeck
- garnaat
- Nathaniel Moseley
- Bradley Ayers
- jibs
- Kenneth Falck
- chirag
- Sean O'Connor

- Scott Moser
- Vineeth Pillai
- Greg Taylor
- root
- darktable
- flipkin
- brimcfadden
- Samuel Lucidi
- Terence Honles
- Mike Schwartz
- Waldemar Kornewald
- Lucas Hrabovsky
- thaDude
- Vinicius Ruan Cainelli
- David Marin
- Stanislav Ievlev
- Victor Trac
- Dan Fairs
- David Pisoni
- Matt Robenolt
- Matt Billenstein
- rgrp
- vikalp
- Christoph Kern
- Gabriel Monroy
- Ben Burry
- Hinnerk
- Jann Kleen
- Louis R. Marascio
- Matt Singleton
- David Park
- Nick Tarleton
- Cory Mintz
- Robert Mela
- rlotun
- John Walsh

- Keith Fitzgerald
- Pierre Riteau
- ryancustommade
- Fabian Topfstedt
- Michael Thompson
- sanbornm
- Seth Golub
- Jon Colverson
- Steve Howard
- Roberto Gaiser
- James Downs
- Gleicon Moraes
- Blake Maltby
- Mac Morgan
- Rytis Sileika
- winhamwr

4.77 Major changes for release 2.0b1

- Support for versioning in S3
- Support for MFA Delete in S3
- Support for Elastic Map Reduce
- Support for Simple Notification Service
- Support for Google Storage
- Support for Consistent Reads and Conditional Puts in SimpleDB
- Significant updates and improvements to Mechanical Turk (mturk) module
- Support for Windows Bundle Tasks in EC2
- Support for Reduced Redundancy Storage (RRS) in S3
- Support for Cluster Computing instances and Placement Groups in EC2

4.78 Getting Started with Boto

This tutorial will walk you through installing and configuring `boto`, as well how to use it to make API calls.

This tutorial assumes you are familiar with Python & that you have registered for an [Amazon Web Services](#) account. You'll need retrieve your `Access Key ID` and `Secret Access Key` from the web-based console.

4.78.1 Installing Boto

You can use `pip` to install the latest released version of `boto`:

```
pip install boto
```

If you want to install `boto` from source:

```
git clone git://github.com/boto/boto.git
cd boto
python setup.py install
```

Note: For most services, this is enough to get going. However, to support everything Boto ships with, you should additionally run `pip install -r requirements.txt`.

This installs all additional, non-stdlib modules, enabling use of things like `boto.cloudsearch`, `boto.manage` & `boto.mashups`, as well as covering everything needed for the test suite.

4.78.2 Using Virtual Environments

Another common way to install `boto` is to use a `virtualenv`, which provides isolated environments. First, install the `virtualenv` Python package:

```
pip install virtualenv
```

Next, create a virtual environment by using the `virtualenv` command and specifying where you want the `virtualenv` to be created (you can specify any directory you like, though this example allows for compatibility with `virtualenvwrapper`):

```
mkdir ~/.virtualenvs
virtualenv ~/.virtualenvs/boto
```

You can now activate the virtual environment:

```
source ~/.virtualenvs/boto/bin/activate
```

Now, any usage of `python` or `pip` (within the current shell) will default to the new, isolated version within your `virtualenv`.

You can now install `boto` into this virtual environment:

```
pip install boto
```

When you are done using `boto`, you can deactivate your virtual environment:

```
deactivate
```

If you are creating a lot of virtual environments, `virtualenvwrapper` is an excellent tool that lets you easily manage your virtual environments.

4.78.3 Configuring Boto Credentials

You have a few options for configuring `boto` (see *Boto Config*). For this tutorial, we'll be using a configuration file. First, create a `~/.boto` file with these contents:

```
[Credentials]
aws_access_key_id = YOURACCESSKEY
aws_secret_access_key = YOURSECRETKEY
```

boto supports a number of configuration values. For more information, see *Boto Config*. The above file, however, is all we need for now. You're now ready to use boto.

4.78.4 Making Connections

boto provides a number of convenience functions to simplify connecting to a service. For example, to work with S3, you can run:

```
>>> import boto
>>> s3 = boto.connect_s3()
```

If you want to connect to a different region, you can import the service module and use the `connect_to_region` functions. For example, to create an EC2 client in 'us-west-2' region, you'd run the following:

```
>>> import boto.ec2
>>> ec2 = boto.ec2.connect_to_region('us-west-2')
```

4.78.5 Troubleshooting Connections

When calling the various `connect_*` functions, you might run into an error like this:

```
>>> import boto
>>> s3 = boto.connect_s3()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "boto/__init__.py", line 121, in connect_s3
    return S3Connection(aws_access_key_id, aws_secret_access_key, **kwargs)
  File "boto/s3/connection.py", line 171, in __init__
    validate_certs=validate_certs)
  File "boto/connection.py", line 548, in __init__
    host, config, self.provider, self._required_auth_capability())
  File "boto/auth.py", line 668, in get_auth_handler
    'Check your credentials' % (len(names), str(names)))
boto.exception.NoAuthHandlerFound: No handler was ready to authenticate. 1 handlers_
↳ were checked. ['HmacAuthV1Handler'] Check your credentials
```

This is because boto cannot find credentials to use. Verify that you have created a `~/.boto` file as shown above. You can also turn on debug logging to verify where your credentials are coming from:

```
>>> import boto
>>> boto.set_stream_logger('boto')
>>> s3 = boto.connect_s3()
2012-12-10 17:15:03,799 boto [DEBUG]:Using access key found in config file.
2012-12-10 17:15:03,799 boto [DEBUG]:Using secret key found in config file.
```

4.78.6 Interacting with AWS Services

Once you have a client for the specific service you want, there are methods on that object that will invoke API operations for that service. The following code demonstrates how to create a bucket and put an object in that bucket:

```
>>> import boto
>>> import time
>>> s3 = boto.connect_s3()

# Create a new bucket. Buckets must have a globally unique name (not just
# unique to your account).
>>> bucket = s3.create_bucket('boto-demo-%s' % int(time.time()))

# Create a new key/value pair.
>>> key = bucket.new_key('mykey')
>>> key.set_contents_from_string("Hello World!")

# Sleep to ensure the data is eventually there.
>>> time.sleep(2)

# Retrieve the contents of ``mykey``.
>>> print key.get_contents_as_string()
'Hello World!'

# Delete the key.
>>> key.delete()
# Delete the bucket.
>>> bucket.delete()
```

Each service supports a different set of commands. You'll want to refer to the other guides & API references in this documentation, as well as referring to the [official AWS API documentation](#).

4.78.7 Next Steps

For many of the services that `boto` supports, there are tutorials as well as detailed API documentation. If you are interested in a specific service, the tutorial for the service is a good starting point. For instance, if you'd like more information on S3, check out the *S3 Tutorial* and the *S3 API reference*.

4.79 An Introduction to boto's EC2 interface

This tutorial focuses on the `boto` interface to the Elastic Compute Cloud from Amazon Web Services. This tutorial assumes that you have already downloaded and installed `boto`.

4.79.1 Creating a Connection

The first step in accessing EC2 is to create a connection to the service. The recommended way of doing this in `boto` is:

```
>>> import boto.ec2
>>> conn = boto.ec2.connect_to_region("us-west-2",
...     aws_access_key_id='<aws access key>',
...     aws_secret_access_key='<aws secret key>')
```

At this point the variable `conn` will point to an `EC2Connection` object. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the `boto` config environment variables and then simply specify which region you want as follows:

```
>>> conn = boto.ec2.connect_to_region("us-west-2")
```

In either case, `conn` will point to an `EC2Connection` object which we will use throughout the remainder of this tutorial.

4.79.2 Launching Instances

Possibly, the most important and common task you'll use EC2 for is to launch, stop and terminate instances. In its most primitive form, you can launch an instance as follows:

```
>>> conn.run_instances('<ami-image-id>')
```

This will launch an instance in the specified region with the default parameters. You will not be able to SSH into this machine, as it doesn't have a security group set. See [EC2 Security Groups](#) for details on creating one.

Now, let's say that you already have a key pair, want a specific type of instance, and you have your *security group* all setup. In this case we can use the keyword arguments to accomplish that:

```
>>> conn.run_instances(
    '<ami-image-id>',
    key_name='myKey',
    instance_type='c1.xlarge',
    security_groups=['your-security-group-here'])
```

The main caveat with the above call is that it is possible to request an instance type that is not compatible with the provided AMI (for example, the instance was created for a 64-bit instance and you choose a `m1.small` `instance_type`). For more details on the plethora of possible keyword parameters, be sure to check out boto's [EC2 API reference](#).

4.79.3 Stopping Instances

Once you have your instances up and running, you might wish to shut them down if they're not in use. Please note that this will only de-allocate virtual hardware resources (as well as instance store drives), but won't destroy your EBS volumes – this means you'll pay nominal provisioned EBS storage fees even if your instance is stopped. To do this, you can do so as follows:

```
>>> conn.stop_instances(instance_ids=['instance-id-1', 'instance-id-2', ...])
```

This will request a 'graceful' stop of each of the specified instances. If you wish to request the equivalent of unplugging your instance(s), simply add `force=True` keyword argument to the call above. Please note that stop instance is not allowed with Spot instances.

4.79.4 Terminating Instances

Once you are completely done with your instance and wish to surrender both virtual hardware, root EBS volume and all other underlying components you can request instance termination. To do so you can use the call below:

```
>>> conn.terminate_instances(instance_ids=['instance-id-1', 'instance-id-2', ...])
```

Please use with care since once you request termination for an instance there is no turning back.

4.79.5 Checking What Instances Are Running

You can also get information on your currently running instances:

```
>>> reservations = conn.get_all_reservations()
>>> reservations
[Reservation:r-00000000]
```

A reservation corresponds to a command to start instances. You can see what instances are associated with a reservation:

```
>>> instances = reservations[0].instances
>>> instances
[Instance:i-00000000]
```

An instance object allows you get more meta-data available about the instance:

```
>>> inst = instances[0]
>>> inst.instance_type
u'c1.xlarge'
>>> inst.placement
u'us-west-2'
```

In this case, we can see that our instance is a `c1.xlarge` instance in the `us-west-2` availability zone.

4.79.6 Checking Health Status Of Instances

You can also get the health status of your instances, including any scheduled events:

```
>>> statuses = conn.get_all_instance_status()
>>> statuses
[InstanceStatus:i-00000000]
```

An instance status object allows you to get information about impaired functionality or scheduled / system maintenance events:

```
>>> status = statuses[0]
>>> status.events
[Event:instance-reboot]
>>> event = status.events[0]
>>> event.description
u'Maintenance software update.'
>>> event.not_before
u'2011-12-11T04:00:00.000Z'
>>> event.not_after
u'2011-12-11T10:00:00.000Z'
>>> status.instance_status
Status:ok
>>> status.system_status
Status:ok
>>> status.system_status.details
{'u'reachability': u'passed'}
```

This will by default include the health status only for running instances. If you wish to request the health status for all instances, simply add `include_all_instances=True` keyword argument to the call above.

4.80 Using Elastic Block Storage (EBS)

4.80.1 EBS Basics

EBS can be used by EC2 instances for permanent storage. Note that EBS volumes must be in the same availability zone as the EC2 instance you wish to attach it to.

To actually create a volume you will need to specify a few details. The following example will create a 50GB EBS in one of the *us-west-2* availability zones:

```
>>> vol = conn.create_volume(50, "us-west-2")
>>> vol
Volume:vol-00000000
```

You can check that the volume is now ready and available:

```
>>> curr_vol = conn.get_all_volumes([vol.id])[0]
>>> curr_vol.status
u'available'
>>> curr_vol.zone
u'us-west-2'
```

We can now attach this volume to the EC2 instance we created earlier, making it available as a new device:

```
>>> conn.attach_volume(vol.id, inst.id, "/dev/sdx")
u'attaching'
```

You will now have a new volume attached to your instance. Note that with some Linux kernels, */dev/sdx* may get translated to */dev/xvdx*. This device can now be used as a normal block device within Linux.

4.80.2 Working With Snapshots

Snapshots allow you to make point-in-time snapshots of an EBS volume for future recovery. Snapshots allow you to create incremental backups, and can also be used to instantiate multiple new volumes. Snapshots can also be used to move EBS volumes across availability zones or making backups to S3.

Creating a snapshot is easy:

```
>>> snapshot = conn.create_snapshot(vol.id, 'My snapshot')
>>> snapshot
Snapshot:snap-00000000
```

Once you have a snapshot, you can create a new volume from it. Volumes are created lazily from snapshots, which means you can start using such a volume straight away:

```
>>> new_vol = snapshot.create_volume('us-west-2')
>>> conn.attach_volume(new_vol.id, inst.id, "/dev/sdy")
u'attaching'
```

If you no longer need a snapshot, you can also easily delete it:

```
>>> conn.delete_snapshot(snapshot.id)
True
```

4.80.3 Working With Launch Configurations

Launch Configurations allow you to create a re-usable set of properties for an instance. These are used with AutoScaling groups to produce consistent repeatable instances sets.

Creating a Launch Configuration is easy:

```
>>> conn = boto.connect_autoscale()
>>> config = LaunchConfiguration(name='foo', image_id='ami-abcd1234', key_name='foo.
↳pem')
>>> conn.create_launch_configuration(config)
```

Once you have a launch configuration, you can list you current configurations:

```
>>> conn = boto.connect_autoscale()
>>> config = conn.get_all_launch_configurations(names=['foo'])
```

If you no longer need a launch configuration, you can delete it:

```
>>> conn = boto.connect_autoscale()
>>> conn.delete_launch_configuration('foo')
```

Changed in version 2.27.0.

Note: If `use_block_device_types=True` is passed to the connection it will deserialize Launch Configurations with Block Device Mappings into a re-usable format with `BlockDeviceType` objects, similar to how AMIs are deserialized currently. Legacy behavior is to put them into a format that is incompatible with creating new Launch Configurations. This switch is in place to preserve backwards compatability, but its usage is the preferred format going forward.

If you would like to use the new format, you should use something like:

```
>>> conn = boto.connect_autoscale(use_block_device_types=True)
>>> config = conn.get_all_launch_configurations(names=['foo'])
```

4.81 EC2 Security Groups

Amazon defines a security group as:

“A security group is a named collection of access rules. These access rules specify which ingress, i.e. incoming, network traffic should be delivered to your instance.”

To get a listing of all currently defined security groups:

```
>>> rs = conn.get_all_security_groups()
>>> print rs
[SecurityGroup:appserver, SecurityGroup:default, SecurityGroup:vnc,
↳SecurityGroup:webserver]
```

Each security group can have an arbitrary number of rules which represent different network ports which are being enabled. To find the rules for a particular security group, use the `rules` attribute:

```
>>> sg = rs[1]
>>> sg.name
u'default'
>>> sg.rules
[IPPermissions:tcp(0-65535),
 IPPermissions:udp(0-65535),
 IPPermissions:icmp(-1--1),
 IPPermissions:tcp(22-22),
 IPPermissions:tcp(80-80)]
```

In addition to listing the available security groups you can also create a new security group. I'll follow through the “Three Tier Web Service” example included in the EC2 Developer's Guide for an example of how to create security groups and add rules to them.

First, let's create a group for our Apache web servers that allows HTTP access to the world:

```
>>> web = conn.create_security_group('apache', 'Our Apache Group')
>>> web
SecurityGroup:apache
>>> web.authorize('tcp', 80, 80, '0.0.0.0/0')
True
```

The first argument is the ip protocol which can be one of; tcp, udp or icmp. The second argument is the FromPort or the beginning port in the range, the third argument is the ToPort or the ending port in the range and the last argument is the CIDR IP range to authorize access to.

Next we create another group for the app servers:

```
>>> app = conn.create_security_group('appserver', 'The application tier')
```

We then want to grant access between the web server group and the app server group. So, rather than specifying an IP address as we did in the last example, this time we will specify another SecurityGroup object.:

```
>>> app.authorize(src_group=web)
True
```

Now, to verify that the web group now has access to the app servers, we want to temporarily allow SSH access to the web servers from our computer. Let's say that our IP address is 192.168.1.130 as it is in the EC2 Developer Guide. To enable that access:

```
>>> web.authorize(ip_protocol='tcp', from_port=22, to_port=22, cidr_ip='192.168.1.130/
↪32')
True
```

Now that this access is authorized, we could ssh into an instance running in the web group and then try to telnet to specific ports on servers in the appserver group, as shown in the EC2 Developer's Guide. When this testing is complete, we would want to revoke SSH access to the web server group, like this:

```
>>> web.rules
[IPPermissions:tcp(80-80),
 IPPermissions:tcp(22-22)]
>>> web.revoke('tcp', 22, 22, cidr_ip='192.168.1.130/32')
True
>>> web.rules
[IPPermissions:tcp(80-80)]
```

4.82 An Introduction to boto's Elastic Mapreduce interface

This tutorial focuses on the boto interface to Elastic Mapreduce from Amazon Web Services. This tutorial assumes that you have already downloaded and installed boto.

4.82.1 Creating a Connection

The first step in accessing Elastic Mapreduce is to create a connection to the service. There are two ways to do this in boto. The first is:

```
>>> from boto.emr.connection import EmrConnection
>>> conn = EmrConnection('<aws access key>', '<aws secret key>')
```

At this point the variable `conn` will point to an `EmrConnection` object. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

`AWS_ACCESS_KEY_ID` - Your AWS Access Key ID `AWS_SECRET_ACCESS_KEY` - Your AWS Secret Access Key

and then call the constructor without any arguments, like this:

```
>>> conn = EmrConnection()
```

There is also a shortcut function in boto that makes it easy to create EMR connections:

```
>>> import boto.emr
>>> conn = boto.emr.connect_to_region('us-west-2')
```

In either case, `conn` points to an `EmrConnection` object which we will use throughout the remainder of this tutorial.

4.82.2 Creating Streaming JobFlow Steps

Upon creating a connection to Elastic Mapreduce you will next want to create one or more jobflow steps. There are two types of steps, streaming and custom jar, both of which have a class in the boto Elastic Mapreduce implementation.

Creating a streaming step that runs the AWS wordcount example, itself written in Python, can be accomplished by:

```
>>> from boto.emr.step import StreamingStep
>>> step = StreamingStep(name='My wordcount example',
...                      mapper='s3n://elasticmapreduce/samples/wordcount/
↳wordSplitter.py',
...                      reducer='aggregate',
...                      input='s3n://elasticmapreduce/samples/wordcount/input',
...                      output='s3n://<my output bucket>/output/wordcount_output')
```

where `<my output bucket>` is a bucket you have created in S3.

Note that this statement does not run the step, that is accomplished later when we create a jobflow.

Additional arguments of note to the streaming jobflow step are `cache_files`, `cache_archive` and `step_args`. The options `cache_files` and `cache_archive` enable you to use the Hadoops distributed cache to share files amongst the instances that run the step. The argument `step_args` allows one to pass additional arguments to Hadoop streaming, for example modifications to the Hadoop job configuration.

4.82.3 Creating Custom Jar Job Flow Steps

The second type of jobflow step executes tasks written with a custom jar. Creating a custom jar step for the AWS CloudBurst example can be accomplished by:

```
>>> from boto.emr.step import JarStep
>>> step = JarStep(name='Coudburst example',
...               jar='s3n://elasticmapreduce/samples/cloudburst/cloudburst.jar',
...               step_args=['s3n://elasticmapreduce/samples/cloudburst/input/s_suis.
↳br',
...                           's3n://elasticmapreduce/samples/cloudburst/input/100k.br
↳',
...                           's3n://<my output bucket>/output/cloudfront_output',
...                           36, 3, 0, 1, 240, 48, 24, 24, 128, 16])
```

Note that this statement does not actually run the step, that is accomplished later when we create a jobflow. Also note that this JarStep does not include a main_class argument since the jar MANIFEST.MF has a Main-Class entry.

4.82.4 Creating JobFlows

Once you have created one or more jobflow steps, you will next want to create and run a jobflow. Creating a jobflow that executes either of the steps we created above can be accomplished by:

```
>>> import boto.emr
>>> conn = boto.emr.connect_to_region('us-west-2')
>>> jobid = conn.run_jobflow(name='My jobflow',
...                          log_uri='s3://<my log uri>/jobflow_logs',
...                          steps=[step])
```

The method will not block for the completion of the jobflow, but will immediately return. The status of the jobflow can be determined by:

```
>>> status = conn.describe_jobflow(jobid)
>>> status.state
u'STARTING'
```

One can then use this state to block for a jobflow to complete. Valid jobflow states currently defined in the AWS API are COMPLETED, FAILED, TERMINATED, RUNNING, SHUTTING_DOWN, STARTING and WAITING.

In some cases you may not have built all of the steps prior to running the jobflow. In these cases additional steps can be added to a jobflow by running:

```
>>> conn.add_jobflow_steps(jobid, [second_step])
```

If you wish to add additional steps to a running jobflow you may want to set the keep_alive parameter to True in run_jobflow so that the jobflow does not automatically terminate when the first step completes.

The run_jobflow method has a number of important parameters that are worth investigating. They include parameters to change the number and type of EC2 instances on which the jobflow is executed, set a SSH key for manual debugging and enable AWS console debugging.

4.82.5 Terminating JobFlows

By default when all the steps of a jobflow have finished or failed the jobflow terminates. However, if you set the keep_alive parameter to True or just want to halt the execution of a jobflow early you can terminate a jobflow by:

```
>>> import boto.emr
>>> conn = boto.emr.connect_to_region('us-west-2')
>>> conn.terminate_jobflow('<jobflow id>')
```

4.83 An Introduction to boto's Autoscale interface

This tutorial focuses on the boto interface to the Autoscale service. This assumes you are familiar with boto's EC2 interface and concepts.

4.83.1 Autoscale Concepts

The AWS Autoscale service is comprised of three core concepts:

1. *Autoscale Group (AG)*: An AG can be viewed as a collection of criteria for maintaining or scaling a set of EC2 instances over one or more availability zones. An AG is limited to a single region.
2. *Launch Configuration (LC)*: An LC is the set of information needed by the AG to launch new instances - this can encompass image ids, startup data, security groups and keys. Only one LC is attached to an AG.
3. *Triggers*: A trigger is essentially a set of rules for determining when to scale an AG up or down. These rules can encompass a set of metrics such as average CPU usage across instances, or incoming requests, a threshold for when an action will take place, as well as parameters to control how long to wait after a threshold is crossed.

4.83.2 Creating a Connection

The first step in accessing autoscaling is to create a connection to the service. There are two ways to do this in boto. The first is:

```
>>> from boto.ec2.autoscale import AutoScaleConnection
>>> conn = AutoScaleConnection('<aws access key>', '<aws secret key>')
```

A Note About Regions and Endpoints

Like EC2 the Autoscale service has a different endpoint for each region. By default the US endpoint is used. To choose a specific region, instantiate the AutoScaleConnection object with that region's endpoint.

```
>>> import boto.ec2.autoscale
>>> autoscale = boto.ec2.autoscale.connect_to_region('eu-west-1')
```

Alternatively, edit your boto.cfg with the default Autoscale endpoint to use:

```
[Boto]
autoscale_endpoint = autoscaling.eu-west-1.amazonaws.com
```

Getting Existing AutoScale Groups

To retrieve existing autoscale groups:

```
>>> conn.get_all_groups()
```

You will get back a list of AutoScale group objects, one for each AG you have.

4.83.3 Creating Autoscaling Groups

An Autoscaling group has a number of parameters associated with it.

1. *Name*: The name of the AG.
2. *Availability Zones*: The list of availability zones it is defined over.
3. *Minimum Size*: Minimum number of instances running at one time.
4. *Maximum Size*: Maximum number of instances running at one time.
5. *Launch Configuration (LC)*: A set of instructions on how to launch an instance.
6. *Load Balancer*: An optional ELB load balancer to use. See the ELB tutorial for information on how to create a load balancer.

For the purposes of this tutorial, let's assume we want to create one autoscale group over the us-east-1a and us-east-1b availability zones. We want to have two instances in each availability zone, thus a minimum size of 4. For now we won't worry about scaling up or down - we'll introduce that later when we talk about triggers. Thus we'll set a maximum size of 4 as well. We'll also associate the AG with a load balancer which we assume we've already created, called 'my_lb'.

Our LC tells us how to start an instance. This will at least include the image id to use, security_group, and key information. We assume the image id, key name and security groups have already been defined elsewhere - see the EC2 tutorial for information on how to create these.

```
>>> from boto.ec2.autoscale import LaunchConfiguration
>>> from boto.ec2.autoscale import AutoScalingGroup
>>> lc = LaunchConfiguration(name='my-launch-config', image_id='my-ami',
                             key_name='my_key_name',
                             security_groups=['my_security_groups'])
>>> conn.create_launch_configuration(lc)
```

We now have created a launch configuration called 'my-launch-config'. We are now ready to associate it with our new autoscale group.

```
>>> ag = AutoScalingGroup(group_name='my_group', load_balancers=['my-lb'],
                           availability_zones=['us-east-1a', 'us-east-1b'],
                           launch_config=lc, min_size=4, max_size=8,
                           connection=conn)
>>> conn.create_auto_scaling_group(ag)
```

We now have a new autoscaling group defined! At this point instances should be starting to launch. To view activity on an autoscale group:

```
>>> ag.get_activities()
[Activity:Launching a new EC2 instance status:Successful progress:100,
 ...]
```

or alternatively:

```
>>> conn.get_all_activities(ag)
```

This autoscale group is fairly useful in that it will maintain the minimum size without breaching the maximum size defined. That means if one instance crashes, the autoscale group will use the launch configuration to start a new one in an attempt to maintain its minimum defined size. It knows instance health using the health check defined on its associated load balancer.

Scaling a Group Up or Down

It can also be useful to scale a group up or down depending on certain criteria. For example, if the average CPU utilization of the group goes above 70%, you may want to scale up the number of instances to deal with demand. Likewise, you might want to scale down if usage drops again. These rules for **how** to scale are defined by *Scaling Policies*, and the rules for **when** to scale are defined by *CloudWatch Metric Alarms*.

For example, let's configure scaling for the above group based on CPU utilization. We'll say it should scale up if the average CPU usage goes above 70% and scale down if it goes below 40%.

Firstly, define some Scaling Policies. These tell Auto Scaling how to scale the group (but not when to do it, we'll specify that later).

We need one policy for scaling up and one for scaling down.

```
>>> from boto.ec2.autoscale import ScalingPolicy
>>> scale_up_policy = ScalingPolicy(
    name='scale_up', adjustment_type='ChangeInCapacity',
    as_name='my_group', scaling_adjustment=1, cooldown=180)
>>> scale_down_policy = ScalingPolicy(
    name='scale_down', adjustment_type='ChangeInCapacity',
    as_name='my_group', scaling_adjustment=-1, cooldown=180)
```

The policy objects are now defined locally. Let's submit them to AWS.

```
>>> conn.create_scaling_policy(scale_up_policy)
>>> conn.create_scaling_policy(scale_down_policy)
```

Now that the policies have been digested by AWS, they have extra properties that we aren't aware of locally. We need to refresh them by requesting them back again.

```
>>> scale_up_policy = conn.get_all_policies(
    as_group='my_group', policy_names=['scale_up'])[0]
>>> scale_down_policy = conn.get_all_policies(
    as_group='my_group', policy_names=['scale_down'])[0]
```

Specifically, we'll need the Amazon Resource Name (ARN) of each policy, which will now be a property of our *ScalingPolicy* objects.

Next we'll create CloudWatch alarms that will define when to run the Auto Scaling Policies.

```
>>> import boto.ec2.cloudwatch
>>> cloudwatch = boto.ec2.cloudwatch.connect_to_region('us-west-2')
```

It makes sense to measure the average CPU usage across the whole Auto Scaling Group, rather than individual instances. We express that as CloudWatch *Dimensions*.

```
>>> alarm_dimensions = {"AutoScalingGroupName": 'my_group'}
```

Create an alarm for when to scale up, and one for when to scale down.

```
>>> from boto.ec2.cloudwatch import MetricAlarm
>>> scale_up_alarm = MetricAlarm(
    name='scale_up_on_cpu', namespace='AWS/EC2',
    metric='CPUUtilization', statistic='Average',
    comparison='>', threshold='70',
    period='60', evaluation_periods=2,
    alarm_actions=[scale_up_policy.policy_arn],
```

(continues on next page)

(continued from previous page)

```

        dimensions=alarm_dimensions)
>>> cloudwatch.create_alarm(scale_up_alarm)

```

```

>>> scale_down_alarm = MetricAlarm(
    name='scale_down_on_cpu', namespace='AWS/EC2',
    metric='CPUUtilization', statistic='Average',
    comparison='<', threshold='40',
    period='60', evaluation_periods=2,
    alarm_actions=[scale_down_policy.policy_arn],
    dimensions=alarm_dimensions)
>>> cloudwatch.create_alarm(scale_down_alarm)

```

Auto Scaling will now create a new instance if the existing cluster averages more than 70% CPU for two minutes. Similarly, it will terminate an instance when CPU usage sits below 40%. Auto Scaling will not add or remove instances beyond the limits of the Scaling Group's 'max_size' and 'min_size' properties.

To retrieve the instances in your autoscale group:

```

>>> import boto.ec2
>>> ec2 = boto.ec2.connect_to_region('us-west-2')
>>> group = conn.get_all_groups(names=['my_group'])[0]
>>> instance_ids = [i.instance_id for i in group.instances]
>>> instances = ec2.get_only_instances(instance_ids)

```

To delete your autoscale group, we first need to shutdown all the instances:

```

>>> ag.shutdown_instances()

```

Once the instances have been shutdown, you can delete the autoscale group:

```

>>> ag.delete()

```

You can also delete your launch configuration:

```

>>> lc.delete()

```

4.84 CloudFront

This new boto module provides an interface to Amazon's Content Service, CloudFront.

Warning: This module is not well tested. Paging of distributions is not yet supported. CNAME support is completely untested. Use with caution. Feedback and bug reports are greatly appreciated.

4.84.1 Creating a CloudFront connection

If you've placed your credentials in your \$HOME/.boto config file then you can simply create a CloudFront connection using:

```

>>> import boto
>>> c = boto.connect_cloudfront()

```

If you do not have this file you will need to specify your AWS access key and secret access key:

```
>>> import boto
>>> c = boto.connect_cloudfront('your-aws-access-key-id', 'your-aws-secret-access-key
↳')
```

4.84.2 Working with CloudFront Distributions

Create a new `boto.cloudfront.distribution.Distribution`:

```
>>> origin = boto.cloudfront.origin.S3Origin('mybucket.s3.amazonaws.com')
>>> distro = c.create_distribution(origin=origin, enabled=False, comment='My new_
↳Distribution')
>>> d.domain_name
u'd2oxf3980lnb8l.cloudfront.net'
>>> d.id
u'ECH69M0IW7613'
>>> d.status
u'InProgress'
>>> d.config.comment
u'My new distribution'
>>> d.config.origin
<S3Origin: mybucket.s3.amazonaws.com>
>>> d.config.caller_reference
u'31b8d9cf-a623-4a28-b062-a91856fac6d0'
>>> d.config.enabled
False
```

Note that a new caller reference is created automatically, using `uuid.uuid4()`. The `boto.cloudfront.distribution.Distribution`, `boto.cloudfront.distribution.DistributionConfig` and `boto.cloudfront.distribution.DistributionSummary` objects are defined in the `boto.cloudfront.distribution` module.

To get a listing of all current distributions:

```
>>> rs = c.get_all_distributions()
>>> rs
[<boto.cloudfront.distribution.DistributionSummary instance at 0xe8d4e0>,
 <boto.cloudfront.distribution.DistributionSummary instance at 0xe8d788>]
```

This returns a list of `boto.cloudfront.distribution.DistributionSummary` objects. Note that paging is not yet supported! To get a `boto.cloudfront.distribution.DistributionObject` from a `boto.cloudfront.distribution.DistributionSummary` object:

```
>>> ds = rs[1]
>>> distro = ds.get_distribution()
>>> distro.domain_name
u'd2oxf3980lnb8l.cloudfront.net'
```

To change a property of a distribution object:

```
>>> distro.comment
u'My new distribution'
>>> distro.update(comment='This is a much better comment')
>>> distro.comment
'This is a much better comment'
```

You can also enable/disable a distribution using the following convenience methods:

```
>>> distro.enable() # just calls distro.update(enabled=True)
```

or:

```
>>> distro.disable() # just calls distro.update(enabled=False)
```

The only attributes that can be updated for a Distribution are comment, enabled and cnames.

To delete a `boto.cloudfront.distribution.Distribution`:

```
>>> distro.delete()
```

4.84.3 Invalidating CloudFront Distribution Paths

Invalidate a list of paths in a CloudFront distribution:

```
>>> paths = ['/path/to/file1.html', '/path/to/file2.html', ...]
>>> inval_req = c.create_invalidation_request(u'ECH69MOIW7613', paths)
>>> print inval_req
<InvalidationBatch: IFCT7K03VUETK>
>>> print inval_req.id
u'IFCT7K03VUETK'
>>> print inval_req.paths
[u'/path/to/file1.html', u'/path/to/file2.html', ..]
```

Warning: Each CloudFront invalidation request can only specify up to 1000 paths. If you need to invalidate more than 1000 paths you will need to split up the paths into groups of 1000 or less and create multiple invalidation requests.

This will return a `boto.cloudfront.invalidation.InvalidationBatch` object representing the invalidation request. You can also fetch a single invalidation request for a given distribution using `invalidation_request_status`:

```
>>> inval_req = c.invalidation_request_status(u'ECH69MOIW7613', u'IFCT7K03VUETK')
>>> print inval_req
<InvalidationBatch: IFCT7K03VUETK>
```

The first parameter is the CloudFront distribution id the request belongs to and the second parameter is the invalidation request id.

It's also possible to get *all* invalidations for a given CloudFront distribution:

```
>>> invals = c.get_invalidation_requests(u'ECH69MOIW7613')
>>> print invals
<boto.cloudfront.invalidation.InvalidationListResultSet instance at 0x15d28d0>
```

This will return an instance of `boto.cloudfront.invalidation.InvalidationListResultSet` which is an iterable object that contains a list of `boto.cloudfront.invalidation.InvalidationSummary` objects that describe each invalidation request and its status:

```
>>> for inval in invals:
>>>     print 'Object: %s, ID: %s, Status: %s' % (inval, inval.id, inval.status)
Object: <InvalidationSummary: ICXT2K02SUETK>, ID: ICXT2K02SUETK, Status: Completed
Object: <InvalidationSummary: ITV9SV0PDNY1Y>, ID: ITV9SV0PDNY1Y, Status: Completed
Object: <InvalidationSummary: I1X3F6N0PLGJN5>, ID: I1X3F6N0PLGJN5, Status: Completed
Object: <InvalidationSummary: I1F3G9N0ZLGKN2>, ID: I1F3G9N0ZLGKN2, Status: Completed
...
```

Simply iterating over the `boto.cloudfront.invalidation.InvalidationListResultSet` object will automatically paginate the results on-the-fly as needed by repeatedly requesting more results from CloudFront until there are none left.

If you wish to paginate the results manually you can do so by specifying the `max_items` option when calling `get_invalidation_requests`:

```
>>> invals = c.get_invalidation_requests(u'ECH69MOIW7613', max_items=2)
>>> print len(list(invals))
2
>>> for inval in invals:
>>>     print 'Object: %s, ID: %s, Status: %s' % (inval, inval.id, inval.status)
Object: <InvalidationSummary: ICXT2K02SUETK>, ID: ICXT2K02SUETK, Status: Completed
Object: <InvalidationSummary: ITV9SV0PDNY1Y>, ID: ITV9SV0PDNY1Y, Status: Completed
```

In this case, iterating over the `boto.cloudfront.invalidation.InvalidationListResultSet` object will *only* make a single request to CloudFront and *only* `max_items` invalidation requests are returned by the iterator. To get the next “page” of results pass the `next_marker` attribute of the previous `boto.cloudfront.invalidation.InvalidationListResultSet` object as the `marker` option to the next call to `get_invalidation_requests`:

```
>>> invals = c.get_invalidation_requests(u'ECH69MOIW7613', max_items=10,
↳marker=invals.next_marker)
>>> print len(list(invals))
2
>>> for inval in invals:
>>>     print 'Object: %s, ID: %s, Status: %s' % (inval, inval.id, inval.status)
Object: <InvalidationSummary: I1X3F6N0PLGJN5>, ID: I1X3F6N0PLGJN5, Status: Completed
Object: <InvalidationSummary: I1F3G9N0ZLGKN2>, ID: I1F3G9N0ZLGKN2, Status: Completed
```

You can get the `boto.cloudfront.invalidation.InvalidationBatch` object representing the invalidation request pointed to by a `boto.cloudfront.invalidation.InvalidationSummary` object using:

```
>>> inval_req = inval.get_invalidation_request()
>>> print inval_req
<InvalidationBatch: IFCT7K03VUETK>
```

Similarly you can get the parent `boto.cloudfront.distribution.Distribution` object for the invalidation request from a `boto.cloudfront.invalidation.InvalidationSummary` object using:

```
>>> dist = inval.get_distribution()
>>> print dist
<boto.cloudfront.distribution.Distribution instance at 0x304a7e8>
```

4.85 An Introduction to boto's SimpleDB interface

This tutorial focuses on the boto interface to AWS' SimpleDB. This tutorial assumes that you have boto already downloaded and installed.

Note: If you're starting a new application, you might want to consider using *DynamoDB2* instead, as it has a more comprehensive feature set & has guaranteed performance throughput levels.

4.85.1 Creating a Connection

The first step in accessing SimpleDB is to create a connection to the service. To do so, the most straight forward way is the following:

```
>>> import boto.sdb
>>> conn = boto.sdb.connect_to_region(
...     'us-west-2',
...     aws_access_key_id='<YOUR_AWS_KEY_ID>',
...     aws_secret_access_key='<YOUR_AWS_SECRET_KEY>')
>>> conn
SDBConnection:sdb.amazonaws.com
>>>
```

Bear in mind that if you have your credentials in boto config in your home directory, the two keyword arguments in the call above are not needed. Also important to note is that just as any other AWS service, SimpleDB is region-specific and as such you might want to specify which region to connect to, by default, it'll connect to the US-EAST-1 region.

4.85.2 Creating Domains

Arguably, once you have your connection established, you'll want to create one or more domains. Creating new domains is a fairly straight forward operation. To do so, you can proceed as follows:

```
>>> conn.create_domain('test-domain')
Domain:test-domain
>>>
>>> conn.create_domain('test-domain-2')
Domain:test-domain
>>>
```

Please note that SimpleDB, unlike its newest sibling DynamoDB, is truly and completely schema-less. Thus, there's no need specify domain keys or ranges.

4.85.3 Listing All Domains

Unlike DynamoDB or other database systems, SimpleDB uses the concept of 'domains' instead of tables. So, to list all your domains for your account in a region, you can simply do as follows:

```
>>> domains = conn.get_all_domains()
>>> domains
[Domain:test-domain, Domain:test-domain-2]
>>>
```

The `get_all_domains()` method returns a `boto.resultset.ResultSet` containing all `boto.sdb.domain.Domain` objects associated with this connection's Access Key ID for that region.

4.85.4 Retrieving a Domain (by name)

If you wish to retrieve a specific domain whose name is known, you can do so as follows:

```
>>> dom = conn.get_domain('test-domain')
>>> dom
Domain:test-domain
>>>
```

The `get_domain` call has an optional `validate` parameter, which defaults to `True`. This will make sure to raise an exception if the domain you are looking for doesn't exist. If you set it to `false`, it will return a `Domain` object blindly regardless of its existence.

4.85.5 Getting Domain Metadata

There are times when you might want to know your domains' machine usage, approx. item count and other such data. To this end, boto offers a simple and convenient way to do so as shown below:

```
>>> domain_meta = conn.domain_metadata(dom)
>>> domain_meta
<boto.sdb.domain.DomainMetaData instance at 0x23cd440>
>>> dir(domain_meta)
['BoxUsage', 'DomainMetadataResponse', 'DomainMetadataResult', 'RequestId',
↳ 'ResponseMetadata',
'__doc__', '__init__', '__module__', 'attr_name_count', 'attr_names_size', 'attr_
↳ value_count', 'attr_values_size',
'domain', 'endElement', 'item_count', 'item_names_size', 'startElement', 'timestamp']
>>> domain_meta.item_count
0
>>>
```

Please bear in mind that while in the example above we used a previously retrieved domain object as the parameter, you can retrieve the domain metadata via its name (string).

4.85.6 Adding Items (and attributes)

Once you have your domain setup, presumably, you'll want to start adding items to it. In its most straight forward form, you need to provide a name for the item – think of it as a record id – and a collection of the attributes you want to store in the item (often a Dictionary-like object). So, adding an item to a domain looks as follows:

```
>>> item_name = 'ABC_123'
>>> item_attrs = {'Artist': 'The Jackson 5', 'Genera': 'Pop'}
>>> dom.put_attributes(item_name, item_attrs)
True
>>>
```

Now let's check if it worked:

```
>>> domain_meta = conn.domain_metadata(dom)
>>> domain_meta.item_count
```

(continues on next page)

(continued from previous page)

```
1
>>>
```

4.85.7 Batch Adding Items (and attributes)

You can also add a number of items at the same time in a similar fashion. All you have to provide to the `batch_put_attributes()` method is a Dictionary-like object with your items and their respective attributes, as follows:

```
>>> items = {'item1':{'attr1':'val1'}, 'item2':{'attr2':'val2'}}
>>> dom.batch_put_attributes(items)
True
>>>
```

Now, let's check the item count once again:

```
>>> domain_meta = conn.domain_metadata(dom)
>>> domain_meta.item_count
3
>>>
```

A few words of warning: both `batch_put_attributes()` and `put_item()`, by default, will overwrite the values of the attributes if both the item and attribute already exist. If the item exists, but not the attributes, it will append the new attributes to the attribute list of that item. If you do not wish these methods to behave in that manner, simply supply them with a `'replace=False'` parameter.

4.85.8 Retrieving Items

To retrieve an item along with its attributes is a fairly straight forward operation and can be accomplished as follows:

```
>>> dom.get_item('item1')
{'u'attr1': u'val1'}
>>>
```

Since SimpleDB works in an “eventual consistency” manner, we can also request a forced consistent read (though this will invariably adversely affect read performance). The way to accomplish that is as shown below:

```
>>> dom.get_item('item1', consistent_read=True)
{'u'attr1': u'val1'}
>>>
```

4.85.9 Retrieving One or More Items

Another way to retrieve items is through boto's `select()` method. This method, at the bare minimum, requires a standard SQL select query string and you would do something along the lines of:

```
>>> query = 'select * from `test-domain` where attr1="val1"'
>>> rs = dom.select(query)
>>> for j in rs:
...     print 'o hai'
...
o hai
>>>
```

This method returns a ResultSet collection you can iterate over.

4.85.10 Updating Item Attributes

The easiest way to modify an item's attributes is by manipulating the item's attributes and then saving those changes. For example:

```
>>> item = dom.get_item('item1')
>>> item['attr1'] = 'val_changed'
>>> item.save()
```

4.85.11 Deleting Items (and its attributes)

Deleting an item is a very simple operation. All you are required to provide is either the name of the item or an item object to the `delete_item()` method, boto will take care of the rest:

```
>>> dom.delete_item(item)
>>> True
```

4.85.12 Deleting Domains

To delete a domain and all items under it (i.e. be very careful), you can do it as follows:

```
>>> conn.delete_domain('test-domain')
True
>>>
```

4.86 An Introduction to boto's DynamoDB interface

This tutorial focuses on the boto interface to AWS' [DynamoDB](#). This tutorial assumes that you have boto already downloaded and installed.

Warning: This tutorial covers the **ORIGINAL** release of DynamoDB. It has since been supplanted by a second major version & an updated API to talk to the new version. The documentation for the new version of DynamoDB (& boto's support for it) is at [DynamoDB v2](#).

4.86.1 Creating a Connection

The first step in accessing DynamoDB is to create a connection to the service. To do so, the most straight forward way is the following:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region(
    'us-west-2',
    aws_access_key_id='<YOUR_AWS_KEY_ID>',
    aws_secret_access_key='<YOUR_AWS_SECRET_KEY>')
>>> conn
<boto.dynamodb.layer2.Layer2 object at 0x3fb3090>
```


Bear in mind that if you have your credentials in boto config in your home directory, the two keyword arguments in the call above are not needed. More details on configuration can be found in *Boto Config*.

The `boto.dynamodb.connect_to_region()` function returns a `boto.dynamodb.layer2.Layer2` instance, which is a high-level API for working with DynamoDB. Layer2 is a set of abstractions that sit atop the lower level `boto.dynamodb.layer1.Layer1` API, which closely mirrors the Amazon DynamoDB API. For the purpose of this tutorial, we'll just be covering Layer2.

4.86.2 Listing Tables

Now that we have a DynamoDB connection object, we can then query for a list of existing tables in that region:

```
>>> conn.list_tables()
['test-table', 'another-table']
```

4.86.3 Creating Tables

DynamoDB tables are created with the `Layer2.create_table` method. While DynamoDB's items (a rough equivalent to a relational DB's row) don't have a fixed schema, you do need to create a schema for the table's hash key element, and the optional range key element. This is explained in greater detail in DynamoDB's [Data Model](#) documentation.

We'll start by defining a schema that has a hash key and a range key that are both strings:

```
>>> message_table_schema = conn.create_schema(
    hash_key_name='forum_name',
    hash_key_proto_value=str,
    range_key_name='subject',
    range_key_proto_value=str
)
```

The next few things to determine are table name and read/write throughput. We'll defer explaining throughput to the DynamoDB's [Provisioned Throughput](#) docs.

We're now ready to create the table:

```
>>> table = conn.create_table(
    name='messages',
    schema=message_table_schema,
    read_units=10,
    write_units=10
)
>>> table
Table(messages)
```

This returns a `boto.dynamodb.table.Table` instance, which provides simple ways to create (put), update, and delete items.

4.86.4 Getting a Table

To retrieve an existing table, use `Layer2.get_table`:

```
>>> conn.list_tables()
['test-table', 'another-table', 'messages']
>>> table = conn.get_table('messages')
>>> table
Table(messages)
```

`Layer2.get_table`, like `Layer2.create_table`, returns a `boto.dynamodb.table.Table` instance.

Keep in mind that `Layer2.get_table` will make an API call to retrieve various attributes of the table including the creation time, the read and write capacity, and the table schema. If you already know the schema, you can save an API call and create a `boto.dynamodb.table.Table` object without making any calls to Amazon DynamoDB:

```
>>> table = conn.table_from_schema(
    name='messages',
    schema=message_table_schema)
```

If you do this, the following fields will have `None` values:

- `create_time`
- `status`
- `read_units`
- `write_units`

In addition, the `item_count` and `size_bytes` will be 0. If you create a table object directly from a schema object and decide later that you need to retrieve any of these additional attributes, you can use the `Table.refresh` method:

```
>>> from boto.dynamodb.schema import Schema
>>> table = conn.table_from_schema(
    name='messages',
    schema=Schema.create(hash_key=('forum_name', 'S'),
                        range_key=('subject', 'S')))
>>> print table.write_units
None
>>> # Now we decide we need to know the write_units:
>>> table.refresh()
>>> print table.write_units
10
```

The recommended best practice is to retrieve a table object once and use that object for the duration of your application. So, for example, instead of this:

```
class Application(object):
    def __init__(self, layer2):
        self._layer2 = layer2

    def retrieve_item(self, table_name, key):
        return self._layer2.get_table(table_name).get_item(key)
```

You can do something like this instead:

```
class Application(object):
    def __init__(self, layer2):
        self._layer2 = layer2
        self._tables_by_name = {}
```

(continues on next page)

(continued from previous page)

```

def retrieve_item(self, table_name, key):
    table = self._tables_by_name.get(table_name)
    if table is None:
        table = self._layer2.get_table(table_name)
        self._tables_by_name[table_name] = table
    return table.get_item(key)

```

4.86.5 Describing Tables

To get a complete description of a table, use `Layer2.describe_table`:

```

>>> conn.list_tables()
['test-table', 'another-table', 'messages']
>>> conn.describe_table('messages')
{
  'Table': {
    'CreationDateTime': 1327117581.624,
    'ItemCount': 0,
    'KeySchema': {
      'HashKeyElement': {
        'AttributeName': 'forum_name',
        'AttributeType': 'S'
      },
      'RangeKeyElement': {
        'AttributeName': 'subject',
        'AttributeType': 'S'
      }
    },
    'ProvisionedThroughput': {
      'ReadCapacityUnits': 10,
      'WriteCapacityUnits': 10
    },
    'TableName': 'messages',
    'TableSizeBytes': 0,
    'TableStatus': 'ACTIVE'
  }
}

```

4.86.6 Adding Items

Continuing on with our previously created messages table, adding an:

```

>>> table = conn.get_table('messages')
>>> item_data = {
    'Body': 'http://url_to_lolcat.gif',
    'SentBy': 'User A',
    'ReceivedTime': '12/9/2011 11:36:03 PM',
}
>>> item = table.new_item(
    # Our hash key is 'forum'
    hash_key='LOLCat Forum',
    # Our range key is 'subject'
    range_key='Check this out!',

```

(continues on next page)

(continued from previous page)

```
    # This has the
    attrs=item_data
)
```

The `Table.new_item` method creates a new `boto.dynamodb.item.Item` instance with your specified hash key, range key, and attributes already set. `Item` is a `dict` sub-class, meaning you can edit your data as such:

```
item['a_new_key'] = 'testing'
del item['a_new_key']
```

After you are happy with the contents of the item, use `Item.put` to commit it to DynamoDB:

```
>>> item.put()
```

4.86.7 Retrieving Items

Now, let's check if it got added correctly. Since DynamoDB works under an 'eventual consistency' mode, we need to specify that we wish a consistent read, as follows:

```
>>> table = conn.get_table('messages')
>>> item = table.get_item(
    # Your hash key was 'forum_name'
    hash_key='LOLCat Forum',
    # Your range key was 'subject'
    range_key='Check this out!'
)
>>> item
{
    # Note that this was your hash key attribute (forum_name)
    'forum_name': 'LOLCat Forum',
    # This is your range key attribute (subject)
    'subject': 'Check this out!'
    'Body': 'http://url_to_lolcat.gif',
    'ReceivedTime': '12/9/2011 11:36:03 PM',
    'SentBy': 'User A',
}
```

4.86.8 Updating Items

To update an item's attributes, simply retrieve it, modify the value, then `Item.put` it again:

```
>>> table = conn.get_table('messages')
>>> item = table.get_item(
    hash_key='LOLCat Forum',
    range_key='Check this out!'
)
>>> item['SentBy'] = 'User B'
>>> item.put()
```

4.86.9 Working with Decimals

To avoid the loss of precision, you can stipulate that the `decimal.Decimal` type be used for numeric values:

```
>>> import decimal
>>> conn.use_decimals()
>>> table = conn.get_table('messages')
>>> item = table.new_item(
    hash_key='LOLCat Forum',
    range_key='Check this out!'
)
>>> item['decimal_type'] = decimal.Decimal('1.12345678912345')
>>> item.put()
>>> print table.get_item('LOLCat Forum', 'Check this out!')
{'forum_name': 'LOLCat Forum', u'decimal_type': Decimal('1.12345678912345'),
 u'subject': 'Check this out!'}
```

You can enable the usage of `decimal.Decimal` by using either the `use_decimals` method, or by passing in the `Dynamizer` class for the `dynamizer` param:

```
>>> from boto.dynamodb.types import Dynamizer
>>> conn = boto.dynamodb.connect_to_region(dynamizer=Dynamizer)
```

This mechanism can also be used if you want to customize the encoding/decoding process of DynamoDB types.

4.86.10 Deleting Items

To delete items, use the `Item.delete` method:

```
>>> table = conn.get_table('messages')
>>> item = table.get_item(
    hash_key='LOLCat Forum',
    range_key='Check this out!'
)
>>> item.delete()
```

4.86.11 Deleting Tables

Warning: Deleting a table will also **permanently** delete all of its contents without prompt. Use carefully.

There are two easy ways to delete a table. Through your top-level `Layer2` object:

```
>>> conn.delete_table(table)
```

Or by getting the table, then using `Table.delete`:

```
>>> table = conn.get_table('messages')
>>> table.delete()
```

4.87 An Introduction to boto's RDS interface

This tutorial focuses on the boto interface to the Relational Database Service from Amazon Web Services. This tutorial assumes that you have boto already downloaded and installed, and that you wish to setup a MySQL instance in RDS.

Warning: This tutorial covers the **ORIGINAL** module for RDS. It has since been supplanted by a second major version & an updated API complete with all service operations. The documentation for the new version of boto's support for RDS is at *RDS v2*.

4.87.1 Creating a Connection

The first step in accessing RDS is to create a connection to the service. The recommended method of doing this is as follows:

```
>>> import boto.rds
>>> conn = boto.rds.connect_to_region(
...     "us-west-2",
...     aws_access_key_id='<aws access key>',
...     aws_secret_access_key='<aws secret key>')
```

At this point the variable `conn` will point to an `RDSConnection` object in the US-WEST-2 region. Bear in mind that just as any other AWS service, RDS is region-specific. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

- `AWS_ACCESS_KEY_ID` - Your AWS Access Key ID
- `AWS_SECRET_ACCESS_KEY` - Your AWS Secret Access Key

and then simply call:

```
>>> import boto.rds
>>> conn = boto.rds.connect_to_region("us-west-2")
```

In either case, `conn` will point to an `RDSConnection` object which we will use throughout the remainder of this tutorial.

4.87.2 Starting an RDS Instance

Creating a DB instance is easy. You can do so as follows:

```
>>> db = conn.create_dbinstance("db-master-1", 10, 'db.m1.small', 'root', 'hunter2')
```

This example would create a DB identified as `db-master-1` with 10GB of storage. This instance would be running on `db.m1.small` type, with the login name being `root`, and the password `hunter2`.

To check on the status of your RDS instance, you will have to query the RDS connection again:

```
>>> instances = conn.get_all_dbinstances("db-master-1")
>>> instances
[DBInstance:db-master-1]
>>> db = instances[0]
>>> db.status
u'available'
>>> db.endpoint
(u'db-master-1.aaaaaaaaaa.us-west-2.rds.amazonaws.com', 3306)
```

4.87.3 Creating a Security Group

Before you can actually connect to this RDS service, you must first create a security group. You can add a CIDR range or an *EC2 security group* to your *DB security group*

```
>>> sg = conn.create_dbsecurity_group('web_servers', 'Web front-ends')
>>> sg.authorize(cidr_ip='10.3.2.45/32')
True
```

You can then associate this security group with your RDS instance:

```
>>> db.modify(security_groups=[sg])
```

4.87.4 Connecting to your New Database

Once you have reached this step, you can connect to your RDS instance as you would with any other MySQL instance:

```
>>> db.endpoint
(u'db-master-1.aaaaaaaaa.us-west-2.rds.amazonaws.com', 3306)

% mysql -h db-master-1.aaaaaaaaa.us-west-2.rds.amazonaws.com -u root -phunter2
mysql>
```

4.87.5 Making a backup

You can also create snapshots of your database very easily:

```
>>> db.snapshot('db-master-1-2013-02-05')
DBSnapshot:db-master-1-2013-02-05
```

Once this snapshot is complete, you can create a new database instance from it:

```
>>> db2 = conn.restore_dbinstance_from_dbsnapshot(
...     'db-master-1-2013-02-05',
...     'db-restored-1',
...     'db.m1.small',
...     'us-west-2')
```

4.88 An Introduction to boto's SQS interface

This tutorial focuses on the boto interface to the Simple Queue Service from Amazon Web Services. This tutorial assumes that you have boto already downloaded and installed.

4.88.1 Creating a Connection

The first step in accessing SQS is to create a connection to the service. The recommended method of doing this is as follows:

```
>>> import boto.sqs
>>> conn = boto.sqs.connect_to_region(
...     "us-west-2",
...     aws_access_key_id='<aws access key>',
...     aws_secret_access_key='<aws secret key>')
```

At this point the variable `conn` will point to an `SQSConnection` object in the US-WEST-2 region. Bear in mind that just as any other AWS service, SQS is region-specific. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

- `AWS_ACCESS_KEY_ID` - Your AWS Access Key ID
- `AWS_SECRET_ACCESS_KEY` - Your AWS Secret Access Key

and then simply call:

```
>>> import boto.sqs
>>> conn = boto.sqs.connect_to_region("us-west-2")
```

In either case, `conn` will point to an `SQSConnection` object which we will use throughout the remainder of this tutorial.

4.88.2 Creating a Queue

Once you have a connection established with SQS, you will probably want to create a queue. In its simplest form, that can be accomplished as follows:

```
>>> q = conn.create_queue('myqueue')
```

The `create_queue` method will create (and return) the requested queue if it does not exist or will return the existing queue if it does. There is an optional parameter to `create_queue` called `visibility_timeout`. This basically controls how long a message will remain invisible to other queue readers once it has been read (see SQS documentation for more detailed explanation). If this is not explicitly specified the queue will be created with whatever default value SQS provides (currently 30 seconds). If you would like to specify another value, you could do so like this:

```
>>> q = conn.create_queue('myqueue', 120)
```

This would establish a default visibility timeout for this queue of 120 seconds. As you will see later on, this default value for the queue can also be overridden each time a message is read from the queue. If you want to check what the default visibility timeout is for a queue:

```
>>> q.get_timeout()
30
```

4.88.3 Listing all Queues

To retrieve a list of the queues for your account in the current region:

```
>>> conn.get_all_queues()
[
  Queue(https://queue.amazonaws.com/411358162645/myqueue),
  Queue(https://queue.amazonaws.com/411358162645/another_queue),
  Queue(https://queue.amazonaws.com/411358162645/another_queue2)
]
```

This will leave you with a list of all of your `boto.sqs.queue.Queue` instances. Alternatively, if you wanted to only list the queues that started with 'another':

```
>>> conn.get_all_queues(prefix='another')
[
  Queue(https://queue.amazonaws.com/411358162645/another_queue),
  Queue(https://queue.amazonaws.com/411358162645/another_queue2)
]
```


4.88.4 Getting a Queue (by name)

If you wish to explicitly retrieve an existing queue and the name of the queue is known, you can retrieve the queue as follows:

```
>>> my_queue = conn.get_queue('myqueue')
Queue(https://queue.amazonaws.com/411358162645/myqueue)
```

This leaves you with a single `boto.sqs.queue.Queue`, which abstracts the SQS Queue named 'myqueue'.

4.88.5 Writing Messages

Once you have a queue setup, presumably you will want to write some messages to it. SQS doesn't care what kind of information you store in your messages or what format you use to store it. As long as the amount of data per message is less than or equal to 256Kb, SQS won't complain.

So, first we need to create a Message object:

```
>>> from boto.sqs.message import Message
>>> m = Message()
>>> m.set_body('This is my first message.')
>>> q.write(m)
```

The write method will return the Message object. The `id` and `md5` attribute of the Message object will be updated with the values of the message that was written to the queue.

Arbitrary message attributes can be defined by setting a simple dictionary of values on the message object:

```
>>> m = Message()
>>> m.message_attributes = {
...     "name1": {
...         "data_type": "String",
...         "string_value": "I am a string"
...     },
...     "name2": {
...         "data_type": "Number",
...         "string_value": "12"
...     }
... }
```

Note that by default, these arbitrary attributes are not returned when you request messages from a queue. Instead, you must request them via the `message_attributes` parameter (see below).

If the message cannot be written an `SQSError` exception will be raised.

4.88.6 Writing Messages (Custom Format)

The technique above will work only if you use boto's default Message payload format; however, you may have a lot of specific requirements around the format of the message data. For example, you may want to store one big string or you might want to store something that looks more like RFC822 messages or you might want to store a binary payload such as pickled Python objects.

The way boto deals with this issue is to define a simple Message object that treats the message data as one big string which you can set and get. If that Message object meets your needs, you're good to go. However, if you need to incorporate different behavior in your message or handle different types of data you can create your own Message class. You just need to register that class with the boto queue object so that it knows that, when you read a message

from the queue, it should create one of your message objects rather than the default boto Message object. To register your message class, you would:

```
>>> import MyMessage
>>> q.set_message_class(MyMessage)
>>> m = MyMessage()
>>> m.set_body('This is my first message.')
>>> q.write(m)
```

where MyMessage is the class definition for your message class. Your message class should subclass the boto Message because there is a small bit of Python magic happening in the `__setattr__` method of the boto Message class.

4.88.7 Reading Messages

So, now we have a message in our queue. How would we go about reading it? Here's one way:

```
>>> rs = q.get_messages()
>>> len(rs)
1
>>> m = rs[0]
>>> m.get_body()
u'This is my first message'
```

The `get_messages` method also returns a ResultSet object as described above. In addition to the special attributes that we already talked about the ResultSet object also contains any results returned by the request. To get at the results you can treat the ResultSet as a sequence object (e.g. a list). We can check the length (how many results) and access particular items within the list using the slice notation familiar to Python programmers.

At this point, we have read the message from the queue and SQS will make sure that this message remains invisible to other readers of the queue until the visibility timeout period for the queue expires. If you delete the message before the timeout period expires then no one else will ever see the message again. However, if you don't delete it (maybe because your reader crashed or failed in some way, for example) it will magically reappear in my queue for someone else to read. If you aren't happy with the default visibility timeout defined for the queue, you can override it when you read a message:

```
>>> q.get_messages(visibility_timeout=60)
```

This means that regardless of what the default visibility timeout is for the queue, this message will remain invisible to other readers for 60 seconds.

The `get_messages` method can also return more than a single message. By passing a `num_messages` parameter (defaults to 1) you can control the maximum number of messages that will be returned by the method. To show this feature off, first let's load up a few more messages.

```
>>> for i in range(1, 11):
...     m = Message()
...     m.set_body('This is message %d' % i)
...     q.write(m)
...
>>> rs = q.get_messages(10)
>>> len(rs)
10
```

Don't be alarmed if the length of the result set returned by the `get_messages` call is less than 10. Sometimes it takes some time for new messages to become visible in the queue. Give it a minute or two and they will all show up.

If you want a slightly simpler way to read messages from a queue, you can use the `read` method. It will either return the message read or it will return `None` if no messages were available. You can also pass a `visibility_timeout` parameter to `read`, if you desire:

```
>>> m = q.read(60)
>>> m.get_body()
u'This is my first message'
```

4.88.8 Reading Message Attributes

By default, no arbitrary message attributes are returned when requesting messages. You can change this behavior by specifying the names of attributes you wish to have returned:

```
>>> rs = queue.get_messages(message_attributes=['name1', 'name2'])
>>> print rs[0].message_attributes['name1']['string_value']
```

'I am a string'

A special value of `All` or `.*` may be passed to return all available message attributes.

4.88.9 Deleting Messages and Queues

As stated above, messages are never deleted by the queue unless explicitly told to do so. To remove a message from a queue:

```
>>> q.delete_message(m)
[]
```

If I want to delete the entire queue, I would use:

```
>>> conn.delete_queue(q)
```

This will delete the queue, even if there are still messages within the queue.

4.88.10 Additional Information

The above tutorial covers the basic operations of creating queues, writing messages, reading messages, deleting messages, and deleting queues. There are a few utility methods in boto that might be useful as well. For example, to count the number of messages in a queue:

```
>>> q.count()
10
```

Removing all messages in a queue is as simple as calling `purge`:

```
>>> q.purge()
```

Be REAL careful with that one! Finally, if you want to dump all of the messages in a queue to a local file:

```
>>> q.dump('messages.txt', sep='\n-----\n')
```

This will read all of the messages in the queue and write the bodies of each of the messages to the file `messages.txt`. The optional `sep` argument is a separator that will be printed between each message body in the file.

4.89 Simple Email Service Tutorial

This tutorial focuses on the boto interface to AWS' [Simple Email Service \(SES\)](#). This tutorial assumes that you have boto already downloaded and installed.

4.89.1 Creating a Connection

The first step in accessing SES is to create a connection to the service. To do so, the most straight forward way is the following:

```
>>> import boto.ses
>>> conn = boto.ses.connect_to_region(
    'us-west-2',
    aws_access_key_id='<YOUR_AWS_KEY_ID>',
    aws_secret_access_key='<YOUR_AWS_SECRET_KEY>')
>>> conn
SESConnection:email.us-west-2.amazonaws.com
```

Bear in mind that if you have your credentials in boto config in your home directory, the two keyword arguments in the call above are not needed. More details on configuration can be found in [Boto Config](#).

The `boto.ses.connect_to_region()` functions returns a `boto.ses.connection.SESConnection` instance, which is the boto API for working with SES.

4.89.2 Notes on Sending

It is important to keep in mind that while emails appear to come “from” the address that you specify via Reply-To, the sending is done through Amazon. Some clients do pick up on this disparity, and leave a note on emails.

4.89.3 Verifying a Sender Email Address

Before you can send email “from” an address, you must prove that you have access to the account. When you send a validation request, an email is sent to the address with a link in it. Clicking on the link validates the address and adds it to your SES account. Here’s how to send the validation email:

```
>>> conn.verify_email_address('some@address.com')
{
  'VerifyEmailAddressResponse': {
    'ResponseMetadata': {
      'RequestId': '4a974fd5-56c2-11e1-ad4c-c1f08c91d554'
    }
  }
}
```

After a short amount of time, you’ll find an email with the validation link inside. Click it, and this address may be used to send emails.

4.89.4 Listing Verified Addresses

If you’d like to list the addresses that are currently verified on your SES account, use `list_verified_email_addresses`:

```
>>> conn.list_verified_email_addresses()
{
  'ListVerifiedEmailAddressesResponse': {
    'ListVerifiedEmailAddressesResult': {
      'VerifiedEmailAddresses': [
        'some@address.com',
        'another@address.com'
      ]
    },
    'ResponseMetadata': {
      'RequestId': '2ab45c18-56c3-11e1-be66-ffd2a4549d70'
    }
  }
}
```

4.89.5 Deleting a Verified Address

In the event that you'd like to remove an email address from your account, use `delete_verified_email_address`:

```
>>> conn.delete_verified_email_address('another@address.com')
```

4.89.6 Sending an Email

Sending an email is done via `send_email`:

```
>>> conn.send_email(
    'some@address.com',
    'Your subject',
    'Body here',
    ['recipient-address-1@gmail.com'])
{
  'SendEmailResponse': {
    'ResponseMetadata': {
      'RequestId': '4743c2b7-56c3-11e1-bccd-c99bd68002fd'
    },
    'SendEmailResult': {
      'MessageId': '000001357a177192-7b894025-147a-4705-8455-7c880b0c8270-000000
↪'
    }
  }
}
```

If you're wanting to send a multipart MIME email, see the reference for `send_raw_email`, which is a bit more of a low-level alternative.

4.89.7 Checking your Send Quota

Staying within your quota is critical, since the upper limit is a hard cap. Once you have hit your quota, no further email may be sent until enough time elapses to where your 24 hour email count (rolling continuously) is within acceptable ranges. Use `get_send_quota`:

```
>>> conn.get_send_quota()
{
  'GetSendQuotaResponse': {
    'GetSendQuotaResult': {
      'Max24HourSend': '100000.0',
      'SentLast24Hours': '181.0',
      'MaxSendRate': '28.0'
    },
    'ResponseMetadata': {
      'RequestId': 'u'8a629245-56c4-11e1-9c53-9d5f4d2cc8d3'
    }
  }
}
```

4.89.8 Checking your Send Statistics

In order to fight spammers and ensure quality mail is being sent from SES, Amazon tracks bounces, rejections, and complaints. This is done via `get_send_statistics`. Please be warned that the output is extremely verbose, to the point where we'll just show a short excerpt here:

```
>>> conn.get_send_statistics()
{
  'GetSendStatisticsResponse': {
    'GetSendStatisticsResult': {
      'SendDataPoints': [
        {
          'Complaints': '0',
          'Timestamp': '2012-02-13T05:02:00Z',
          'DeliveryAttempts': '8',
          'Bounces': '0',
          'Rejects': '0'
        },
        {
          'Complaints': '0',
          'Timestamp': '2012-02-13T05:17:00Z',
          'DeliveryAttempts': '12',
          'Bounces': '0',
          'Rejects': '0'
        }
      ]
    }
  }
}
```

4.90 Amazon Simple Workflow Tutorial

This tutorial focuses on boto's interface to AWS SimpleWorkflow service.

4.90.1 What is a workflow?

A workflow is a sequence of multiple activities aimed at accomplishing a well-defined objective. For instance, booking an airline ticket as a workflow may encompass multiple activities, such as selection of itinerary, submission of personal

details, payment validation and booking confirmation.

Except for the start and completion of a workflow, each step has a well-defined predecessor and successor. With that

- on successful completion of an activity the workflow can progress with its execution,
- when one of workflow's activities fails it can be retried,
- and when it keeps failing repeatedly the workflow may regress to the previous step to gather alternative inputs or it may simply fail at that stage.

4.90.2 Why use workflows?

Modelling an application on a workflow provides a useful abstraction layer for writing highly-reliable programs for distributed systems, as individual responsibilities can be delegated to a set of redundant, independent and non-critical processing units.

4.90.3 How does Amazon SWF help you accomplish this?

Amazon SimpleWorkflow service defines an interface for workflow orchestration and provides state persistence for workflow executions.

Amazon SWF applications involve communication between the following entities:

- The Amazon Simple Workflow Service - providing centralized orchestration and workflow state persistence,
- Workflow Executors - some entity starting workflow executions, typically through an action taken by a user or from a cronjob.
- Deciders - a program codifying the business logic, i.e. a set of instructions and decisions. Deciders take decisions based on initial set of conditions and outcomes from activities.
- Activity Workers - their objective is very straightforward: to take inputs, execute the tasks and return a result to the Service.

The Workflow Executor contacts SWF Service and requests instantiation of a workflow. A new workflow is created and its state is stored in the service. The next time a decider contacts SWF service to ask for a decision task, it will be informed about a new workflow execution is taking place and it will be asked to advise SWF service on what the next steps should be. The decider then instructs the service to dispatch specific tasks to activity workers. At the next activity worker poll, the task is dispatched, then executed and the results reported back to the SWF, which then passes them onto the deciders. This exchange keeps happening repeatedly until the decider is satisfied and instructs the service to complete the execution.

4.90.4 Prerequisites

You need a valid access and secret key. The examples below assume that you have exported them to your environment, as follows:

```
bash$ export AWS_ACCESS_KEY_ID=<your access key>
bash$ export AWS_SECRET_ACCESS_KEY=<your secret key>
```

Before workflows and activities can be used, they have to be registered with SWF service:

```
# register.py
import boto.swf.layer2 as swf
from boto.swf.exceptions import SWFTypeAlreadyExistsError, SWFDomainAlreadyExistsError
DOMAIN = 'boto_tutorial'
VERSION = '1.0'

registerables = []
registerables.append(swf.Domain(name=DOMAIN))
for workflow_type in ('HelloWorkflow', 'SerialWorkflow', 'ParallelWorkflow',
↳'SubWorkflow'):
    registerables.append(swf.WorkflowType(domain=DOMAIN, name=workflow_type,
↳version=VERSION, task_list='default'))

for activity_type in ('HelloWorld', 'ActivityA', 'ActivityB', 'ActivityC'):
    registerables.append(swf.ActivityType(domain=DOMAIN, name=activity_type,
↳version=VERSION, task_list='default'))

for swf_entity in registerables:
    try:
        swf_entity.register()
        print swf_entity.name, 'registered successfully'
    except (SWFDomainAlreadyExistsError, SWFTypeAlreadyExistsError):
        print swf_entity.__class__.__name__, swf_entity.name, 'already exists'
```

Execution of the above should produce no errors.

```
bash$ python -i register.py
Domain boto_tutorial already exists
WorkflowType HelloWorld already exists
SerialWorkflow registered successfully
ParallelWorkflow registered successfully
ActivityType HelloWorld already exists
ActivityA registered successfully
ActivityB registered successfully
ActivityC registered successfully
>>>
```

4.90.5 HelloWorld

This example is an implementation of a minimal Hello World workflow. Its execution should unfold as follows:

1. A workflow execution is started.
2. The SWF service schedules the initial decision task.
3. A decider polls for decision tasks and receives one.
4. The decider requests scheduling of an activity task.
5. The SWF service schedules the greeting activity task.
6. An activity worker polls for activity task and receives one.
7. The worker completes the greeting activity.
8. The SWF service schedules a decision task to inform about work outcome.
9. The decider polls and receives a new decision task.
10. The decider schedules workflow completion.

11. The workflow execution finishes.

Workflow logic is encoded in the decider:

```
# hello_decider.py
import boto.swf.layer2 as swf

DOMAIN = 'boto_tutorial'
ACTIVITY = 'HelloWorld'
VERSION = '1.0'
TASKLIST = 'default'

class HelloDecider (swf.Decider):

    domain = DOMAIN
    task_list = TASKLIST
    version = VERSION

    def run(self):
        history = self.poll()
        if 'events' in history:
            # Find workflow events not related to decision scheduling.
            workflow_events = [e for e in history['events']
                               if not e['eventType'].startswith('Decision')]
            last_event = workflow_events[-1]

            decisions = swf.Layer1Decisions()
            if last_event['eventType'] == 'WorkflowExecutionStarted':
                decisions.schedule_activity_task('saying_hi', ACTIVITY, VERSION, task_
↪list=TASKLIST)
            elif last_event['eventType'] == 'ActivityTaskCompleted':
                decisions.complete_workflow_execution()
            self.complete(decisions=decisions)
            return True
```

The activity worker is responsible for printing the greeting message when the activity task is dispatched to it by the service:

```
import boto.swf.layer2 as swf

DOMAIN = 'boto_tutorial'
VERSION = '1.0'
TASKLIST = 'default'

class HelloWorker (swf.ActivityWorker):

    domain = DOMAIN
    version = VERSION
    task_list = TASKLIST

    def run(self):
        activity_task = self.poll()
        if 'activityId' in activity_task:
            print 'Hello, World!'
            self.complete()
            return True
```

With actors implemented we can spin up a workflow execution:

```
$ python
>>> import boto.swf.layer2 as swf
>>> execution = swf.WorkflowType(name='HelloWorkflow', domain='boto_tutorial',
↳version='1.0', task_list='default').start()
>>>
```

From separate terminals run an instance of a worker and a decider to carry out a workflow execution (the worker and decider may run from two independent machines).

```
$ python -i hello_decider.py
>>> while HelloDecider().run(): pass
...
```

```
$ python -i hello_worker.py
>>> while HelloWorker().run(): pass
...
Hello, World!
```

Great. Now, to see what just happened, go back to the original terminal from which the execution was started, and read its history.

```
>>> execution.history()
[{'eventId': 1,
  'eventTimestamp': 1381095173.2539999,
  'eventType': 'WorkflowExecutionStarted',
  'workflowExecutionStartedEventAttributes': {'childPolicy': 'TERMINATE',
                                             'executionStartToCloseTimeout': '3600',
                                             'parentInitiatedEventId': 0,
                                             'taskList': {'name': 'default'},
                                             'taskStartToCloseTimeout': '300',
                                             'workflowType': {'name': 'HelloWorkflow
↳',
                                                           'version': '1.0'}}},
 {'decisionTaskScheduledEventAttributes': {'startToCloseTimeout': '300',
                                           'taskList': {'name': 'default'}},
  'eventId': 2,
  'eventTimestamp': 1381095173.2539999,
  'eventType': 'DecisionTaskScheduled'},
 {'decisionTaskStartedEventAttributes': {'scheduledEventId': 2},
  'eventId': 3,
  'eventTimestamp': 1381095177.5439999,
  'eventType': 'DecisionTaskStarted'},
 {'decisionTaskCompletedEventAttributes': {'scheduledEventId': 2,
                                           'startedEventId': 3},
  'eventId': 4,
  'eventTimestamp': 1381095177.855,
  'eventType': 'DecisionTaskCompleted'},
 {'activityTaskScheduledEventAttributes': {'activityId': 'saying_hi',
                                           'activityType': {'name': 'HelloWorld',
                                                             'version': '1.0'},
                                           'decisionTaskCompletedEventId': 4,
                                           'heartbeatTimeout': '600',
                                           'scheduleToCloseTimeout': '3900',
                                           'scheduleToStartTimeout': '300',
                                           'startToCloseTimeout': '3600',
                                           'taskList': {'name': 'default'}},
  'eventId': 5,
```

(continues on next page)

(continued from previous page)

```

    'eventTimestamp': 1381095177.855,
    'eventType': 'ActivityTaskScheduled'},
{'activityTaskStartedEventAttributes': {'scheduledEventId': 5},
 'eventId': 6,
 'eventTimestamp': 1381095179.427,
 'eventType': 'ActivityTaskStarted'},
{'activityTaskCompletedEventAttributes': {'scheduledEventId': 5,
                                          'startedEventId': 6},
 'eventId': 7,
 'eventTimestamp': 1381095179.6989999,
 'eventType': 'ActivityTaskCompleted'},
{'decisionTaskScheduledEventAttributes': {'startToCloseTimeout': '300',
                                          'taskList': {'name': 'default'}},
 'eventId': 8,
 'eventTimestamp': 1381095179.6989999,
 'eventType': 'DecisionTaskScheduled'},
{'decisionTaskStartedEventAttributes': {'scheduledEventId': 8},
 'eventId': 9,
 'eventTimestamp': 1381095179.7420001,
 'eventType': 'DecisionTaskStarted'},
{'decisionTaskCompletedEventAttributes': {'scheduledEventId': 8,
                                          'startedEventId': 9},
 'eventId': 10,
 'eventTimestamp': 1381095180.026,
 'eventType': 'DecisionTaskCompleted'},
{'eventId': 11,
 'eventTimestamp': 1381095180.026,
 'eventType': 'WorkflowExecutionCompleted',
 'workflowExecutionCompletedEventAttributes': {'decisionTaskCompletedEventId': 10}}]

```

4.90.6 Serial Activity Execution

The following example implements a basic workflow with activities executed one after another.

The business logic, i.e. the serial execution of activities, is encoded in the decider:

```

# serial_decider.py
import time
import boto.swf.layer2 as swf

class SerialDecider(swf.Decider):

    domain = 'boto_tutorial'
    task_list = 'default_tasks'
    version = '1.0'

    def run(self):
        history = self.poll()
        if 'events' in history:
            # Get a list of non-decision events to see what event came in last.
            workflow_events = [e for e in history['events']
                               if not e['eventType'].startswith('Decision')]
            decisions = swf.Layer1Decisions()
            # Record latest non-decision event.
            last_event = workflow_events[-1]

```

(continues on next page)

(continued from previous page)

```

last_event_type = last_event['eventType']
if last_event_type == 'WorkflowExecutionStarted':
    # Schedule the first activity.
    decisions.schedule_activity_task('%s-%i' % ('ActivityA', time.time()),
                                     'ActivityA', self.version, task_list='a_tasks')
elif last_event_type == 'ActivityTaskCompleted':
    # Take decision based on the name of activity that has just completed.
    # 1) Get activity's event id.
    last_event_attrs = last_event['activityTaskCompletedEventAttributes']
    completed_activity_id = last_event_attrs['scheduledEventId'] - 1
    # 2) Extract its name.
    activity_data = history['events'][completed_activity_id]
    activity_attrs = activity_data['activityTaskScheduledEventAttributes']
    activity_name = activity_attrs['activityType']['name']
    # 3) Optionally, get the result from the activity.
    result = last_event['activityTaskCompletedEventAttributes'].get(
↳ 'result')

    # Take the decision.
    if activity_name == 'ActivityA':
        decisions.schedule_activity_task('%s-%i' % ('ActivityB', time.
↳ time()),
                                         'ActivityB', self.version, task_list='b_tasks', input=result)
    if activity_name == 'ActivityB':
        decisions.schedule_activity_task('%s-%i' % ('ActivityC', time.
↳ time()),
                                         'ActivityC', self.version, task_list='c_tasks', input=result)
    elif activity_name == 'ActivityC':
        # Final activity completed. We're done.
        decisions.complete_workflow_execution()

self.complete(decisions=decisions)
return True

```

The workers only need to know which task lists to poll.

```

# serial_worker.py
import time
import boto.swf.layer2 as swf

class MyBaseWorker(swf.ActivityWorker):

    domain = 'boto_tutorial'
    version = '1.0'
    task_list = None

    def run(self):
        activity_task = self.poll()
        if 'activityId' in activity_task:
            # Get input.
            # Get the method for the requested activity.
            try:
                print 'working on activity from tasklist %s at %i' % (self.task_list,
↳ time.time())
                self.activity(activity_task.get('input'))
            except Exception as error:
                self.fail(reason=str(error))

```

(continues on next page)

(continued from previous page)

```

        raise error

        return True

    def activity(self, activity_input):
        raise NotImplementedError

class WorkerA(MyBaseWorker):
    task_list = 'a_tasks'
    def activity(self, activity_input):
        self.complete(result="Now don't be givin him sambuca!")

class WorkerB(MyBaseWorker):
    task_list = 'b_tasks'
    def activity(self, activity_input):
        self.complete()

class WorkerC(MyBaseWorker):
    task_list = 'c_tasks'
    def activity(self, activity_input):
        self.complete()

```

Spin up a workflow execution and run the decider:

```

$ python
>>> import boto.swf.layer2 as swf
>>> execution = swf.WorkflowType(name='SerialWorkflow', domain='boto_tutorial',
↳ version='1.0', task_list='default_tasks').start()
>>>

```

```

$ python -i serial_decider.py
>>> while SerialDecider().run(): pass
...

```

Run the workers. The activities will be executed in order:

```

$ python -i serial_worker.py
>>> while WorkerA().run(): pass
...
working on activity from tasklist a_tasks at 1382046291

```

```

$ python -i serial_worker.py
>>> while WorkerB().run(): pass
...
working on activity from tasklist b_tasks at 1382046541

```

```

$ python -i serial_worker.py
>>> while WorkerC().run(): pass
...
working on activity from tasklist c_tasks at 1382046560

```

Looks good. Now, do the following to inspect the state and history of the execution:

```

>>> execution.describe()
{'executionConfiguration': {'childPolicy': 'TERMINATE',

```

(continues on next page)

(continued from previous page)

```

'executionStartToCloseTimeout': '3600',
'taskList': {'name': 'default_tasks'},
'taskStartToCloseTimeout': '300'},
'executionInfo': {'cancelRequested': False,
'closeStatus': 'COMPLETED',
'closeTimestamp': 1382046560.901,
'execution': {'runId': '12fQ1zSaLmI5+lLXB8ux+8U+hLonnXNZCY9Zy+ZvXgzhE=',
'workflowId': 'SerialWorkflow-1.0-1382046514'}},
'executionStatus': 'CLOSED',
'startTimestamp': 1382046514.994,
'workflowType': {'name': 'SerialWorkflow', 'version': '1.0'}},
'latestActivityTaskTimestamp': 1382046560.632,
'openCounts': {'openActivityTasks': 0,
'openChildWorkflowExecutions': 0,
'openDecisionTasks': 0,
'openTimers': 0}}
>>> execution.history()
...

```

4.90.7 Parallel Activity Execution

When activities are independent from one another, their execution may be scheduled in parallel.

The decider schedules all activities at once and marks progress until all activities are completed, at which point the workflow is completed.

```

# parallel_decider.py

import boto.swf.layer2 as swf
import time

SCHED_COUNT = 5

class ParallelDecider(swf.Decider):

    domain = 'boto_tutorial'
    task_list = 'default'
    def run(self):
        decision_task = self.poll()
        if 'events' in decision_task:
            decisions = swf.Layer1Decisions()
            # Decision* events are irrelevant here and can be ignored.
            workflow_events = [e for e in decision_task['events']
                               if not e['eventType'].startswith('Decision')]
            # Record latest non-decision event.
            last_event = workflow_events[-1]
            last_event_type = last_event['eventType']
            if last_event_type == 'WorkflowExecutionStarted':
                # At start, kickoff SCHED_COUNT activities in parallel.
                for i in range(SCHED_COUNT):
                    decisions.schedule_activity_task('activity%i' % i, 'ActivityA',
↳ '1.0',
                                                    task_list=self.task_list)
            elif last_event_type == 'ActivityTaskCompleted':
                # Monitor progress. When all activities complete, complete workflow.

```

(continues on next page)

(continued from previous page)

```

        completed_count = sum([1 for a in decision_task['events']
                               if a['eventType'] == 'ActivityTaskCompleted'])
    print '%i/%i' % (completed_count, SCHED_COUNT)
    if completed_count == SCHED_COUNT:
        decisions.complete_workflow_execution()
    self.complete(decisions=decisions)
    return True

```

Again, the only bit of information a worker needs is which task list to poll.

```

# parallel_worker.py
import time
import boto.swf.layer2 as swf

class ParallelWorker(swf.ActivityWorker):

    domain = 'boto_tutorial'
    task_list = 'default'

    def run(self):
        """Report current time."""
        activity_task = self.poll()
        if 'activityId' in activity_task:
            print 'working on', activity_task['activityId']
            self.complete(result=str(time.time()))
            return True

```

Spin up a workflow execution and run the decider:

```

$ python -i parallel_decider.py
>>> execution = swf.WorkflowType(name='ParallelWorkflow', domain='boto_tutorial',
↳ version='1.0', task_list='default').start()
>>> while ParallelDecider().run(): pass
...
1/5
2/5
4/5
5/5

```

Run two or more workers to see how the service partitions work execution in parallel.

```

$ python -i parallel_worker.py
>>> while ParallelWorker().run(): pass
...
working on activity1
working on activity3
working on activity4

```

```

$ python -i parallel_worker.py
>>> while ParallelWorker().run(): pass
...
working on activity2
working on activity0

```

As seen above, the work was partitioned between the two running workers.

4.90.8 Sub-Workflows

Sometimes it's desired or necessary to break the process up into multiple workflows.

Since the decider is stateless, it's up to you to determine which workflow is being used and which action you would like to take.

```
import boto.swf.layer2 as swf

class SubWorkflowDecider(swf.Decider):

    domain = 'boto_tutorial'
    task_list = 'default'
    version = '1.0'

    def run(self):
        history = self.poll()
        events = []
        if 'events' in history:
            events = history['events']
            # Collect the entire history if there are enough events to become_
↳paginated
        while 'nextPageToken' in history:
            history = self.poll(next_page_token=history['nextPageToken'])
            if 'events' in history:
                events = events + history['events']

        workflow_type = history['workflowType']['name']

        # Get all of the relevent events that have happened since the last_
↳decision task was started
        workflow_events = [e for e in events
                           if e['eventId'] > history['previousStartedEventId'] and
                           not e['eventType'].startswith('Decision')]

        decisions = swf.Layer1Decisions()

        for event in workflow_events:
            last_event_type = event['eventType']
            if last_event_type == 'WorkflowExecutionStarted':
                if workflow_type == 'SerialWorkflow':
                    decisions.start_child_workflow_execution('SubWorkflow', self.
↳version,
                                                                "subworkflow_1", task_list=self.task_list, input="sub_1")
                elif workflow_type == 'SubWorkflow':
                    for i in range(2):
                        decisions.schedule_activity_task("activity_%d" % i,
↳'ActivityA', self.version, task_list='a_tasks')
                    else:
                        decisions.fail_workflow_execution(reason="Unknown workflow %s
↳" % workflow_type)
                    break

                elif last_event_type == 'ChildWorkflowExecutionCompleted':
                    decisions.schedule_activity_task("activity_2", 'ActivityB', self.
↳version, task_list='b_tasks')

                elif last_event_type == 'ActivityTaskCompleted':
```

(continues on next page)

(continued from previous page)

```

        attrs = event['activityTaskCompletedEventAttributes']
        activity = events[attrs['scheduledEventId'] - 1]
        activity_name = activity['activityTaskScheduledEventAttributes'][
↪ 'activityType']['name']

        if activity_name == 'ActivityA':
            completed_count = sum([1 for a in events if a['eventType'] ==
↪ 'ActivityTaskCompleted'])
            if completed_count == 2:
                # Complete the child workflow
                decisions.complete_workflow_execution()
            elif activity_name == 'ActivityB':
                # Complete the parent workflow
                decisions.complete_workflow_execution()

        self.complete(decisions=decisions)
    return True

```

4.90.9 Misc

Some of these things are not obvious by reading the API documents, so hopefully they help you avoid some time-consuming pitfalls.

Pagination

When the decider polls for new tasks, the maximum number of events it will return at a time is 100 (configurable to a smaller number, but not larger). When running a workflow, this number gets quickly exceeded. If it does, the decision task will contain a key `nextPageToken` which can be submit to the `poll()` call to get the next page of events.

```

decision_task = self.poll()

events = []
if 'events' in decision_task:
    events = decision_task['events']
    while 'nextPageToken' in decision_task:
        decision_task = self.poll(next_page_token=decision_task['nextPageToken'])
        if 'events' in decision_task:
            events += decision_task['events']

```

Depending on your workflow logic, you might not need to aggregate all of the events.

Decision Tasks

When first running deciders and activities, it may seem that the decider gets called for every event that an activity triggers; however, this is not the case. More than one event can happen between decision tasks. The decision task will contain a key `previousStartedEventId` that lets you know the `eventId` of the last `DecisionTaskStarted` event that was processed. Your script will need to handle all of the events that have happened since then, not just the last activity.

```

workflow_events = [e for e in events if e['eventId'] > decision_task[
↪ 'previousStartedEventId']]

```

You may also wish to still filter out tasks that start with 'Decision' or filter it in some other way that fulfills your needs. You will now have to iterate over the `workflow_events` list and respond to each event, as it may contain multiple events.

Filtering Events

When running many tasks in parallel, a common task is searching through the history to see how many events of a particular activity type started, completed, and/or failed. Some basic list comprehension makes this trivial.

```
def filter_completed_events(self, events, type):
    completed = [e for e in events if e['eventType'] == 'ActivityTaskCompleted']
    orig = [events[e['activityTaskCompletedEventAttributes']['scheduledEventId']-1]]
    ↪ for e in completed
    return [e for e in orig if e['activityTaskScheduledEventAttributes']['activityType
    ↪']['name'] == type]
```

4.91 An Introduction to boto's Cloudsearch interface

This tutorial focuses on the boto interface to AWS' Cloudsearch. This tutorial assumes that you have boto already downloaded and installed.

4.91.1 Creating a Connection

The first step in accessing CloudSearch is to create a connection to the service.

The recommended method of doing this is as follows:

```
>>> import boto.cloudsearch
>>> conn = boto.cloudsearch.connect_to_region("us-west-2",
...     aws_access_key_id='<aws access key>',
...     aws_secret_access_key='<aws secret key>')
```

At this point, the variable `conn` will point to a CloudSearch connection object in the `us-west-2` region. Available regions for cloudsearch can be found [here](#). In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

- `AWS_ACCESS_KEY_ID` - Your AWS Access Key ID
- `AWS_SECRET_ACCESS_KEY` - Your AWS Secret Access Key

and then simply call:

```
>>> import boto.cloudsearch
>>> conn = boto.cloudsearch.connect_to_region("us-west-2")
```

In either case, `conn` will point to the Connection object which we will use throughout the remainder of this tutorial.

4.91.2 Creating a Domain

Once you have a connection established with the CloudSearch service, you will want to create a domain. A domain encapsulates the data that you wish to index, as well as indexes and metadata relating to it:

```
>>> from boto.cloudsearch.domain import Domain
>>> domain = Domain(conn, conn.create_domain('demo'))
```

This domain can be used to control access policies, indexes, and the actual document service, which you will use to index and search.

4.91.3 Setting access policies

Before you can connect to a document service, you need to set the correct access properties. For example, if you were connecting from 192.168.1.0, you could give yourself access as follows:

```
>>> our_ip = '192.168.1.0'

>>> # Allow our IP address to access the document and search services
>>> policy = domain.get_access_policies()
>>> policy.allow_search_ip(our_ip)
>>> policy.allow_doc_ip(our_ip)
```

You can use the `allow_search_ip` and `allow_doc_ip` methods to give different CIDR blocks access to searching and the document service respectively.

4.91.4 Creating index fields

Each domain can have up to twenty index fields which are indexed by the CloudSearch service. For each index field, you will need to specify whether it's a text or integer field, as well as optionally a default value:

```
>>> # Create an 'text' index field called 'username'
>>> uname_field = domain.create_index_field('username', 'text')

>>> # Epoch time of when the user last did something
>>> time_field = domain.create_index_field('last_activity',
...                                     'uint',
...                                     default=0)
```

It is also possible to mark an index field as a facet. Doing so allows a search query to return categories into which results can be grouped, or to create drill-down categories:

```
>>> # But it would be neat to drill down into different countries
>>> loc_field = domain.create_index_field('location', 'text', facet=True)
```

Finally, you can also mark a snippet of text as being able to be returned directly in your search query by using the `result` option:

```
>>> # Directly insert user snippets in our results
>>> snippet_field = domain.create_index_field('snippet', 'text', result=True)
```

You can add up to 20 index fields in this manner:

```
>>> follower_field = domain.create_index_field('follower_count',
...                                     'uint',
...                                     default=0)
```

4.91.5 Adding Documents to the Index

Now, we can add some documents to our new search domain. First, you will need a document service object through which queries are sent:

```
>>> doc_service = domain.get_document_service()
```

For this example, we will use a pre-populated list of sample content for our import. You would normally pull such data from your database or another document store:

```
>>> users = [
    {
        'id': 1,
        'username': 'dan',
        'last_activity': 1334252740,
        'follower_count': 20,
        'location': 'USA',
        'snippet': 'Dan likes watching sunsets and rock climbing',
    },
    {
        'id': 2,
        'username': 'dankosaur',
        'last_activity': 1334252904,
        'follower_count': 1,
        'location': 'UK',
        'snippet': 'Likes to dress up as a dinosaur.',
    },
    {
        'id': 3,
        'username': 'danielle',
        'last_activity': 1334252969,
        'follower_count': 100,
        'location': 'DE',
        'snippet': 'Just moved to Germany!'
    },
    {
        'id': 4,
        'username': 'daniella',
        'last_activity': 1334253279,
        'follower_count': 7,
        'location': 'USA',
        'snippet': 'Just like Dan, I like to watch a good sunset, but heights scare_
↪me.',
    }
]
```

When adding documents to our document service, we will batch them together. You can schedule a document to be added by using the `add` method. Whenever you are adding a document, you must provide a unique ID, a version ID, and the actual document to be indexed. In this case, we are using the user ID as our unique ID. The version ID is used to determine which is the latest version of an object to be indexed. If you wish to update a document, you must use a higher version ID. In this case, we are using the time of the user's last activity as a version number:

```
>>> for user in users:
>>>     doc_service.add(user['id'], user['last_activity'], user)
```

When you are ready to send the batched request to the document service, you can do with the `commit` method. Note that cloudsearch will charge per 1000 batch uploads. Each batch upload must be under 5MB:

```
>>> result = doc_service.commit()
```

The result is an instance of `CommitResponse` which will make the plain dictionary response a nice object (ie `result.adds`, `result.deletes`) and raise an exception for us if all of our documents weren't actually committed.

If you wish to use the same document service connection after a commit, you must use `clear_sdf` to clear its internal cache.

4.91.6 Searching Documents

Now, let's try performing a search. First, we will need a `SearchServiceConnection`:

```
>>> search_service = domain.get_search_service()
```

A standard search will return documents which contain the exact words being searched for:

```
>>> results = search_service.search(q="dan")
>>> results.hits
2
>>> map(lambda x: x['id'], results)
[u'1', u'4']
```

The standard search does not look at word order:

```
>>> results = search_service.search(q="dinosaur dress")
>>> results.hits
1
>>> map(lambda x: x['id'], results)
[u'2']
```

It's also possible to do more complex queries using the `bq` argument (Boolean Query). When you are using `bq`, your search terms must be enclosed in single quotes:

```
>>> results = search_service.search(bq="'dan'")
>>> results.hits
2
>>> map(lambda x: x['id'], results)
[u'1', u'4']
```

When you are using boolean queries, it's also possible to use wildcards to extend your search to all words which start with your search terms:

```
>>> results = search_service.search(bq="'dan*')
>>> results.hits
4
>>> map(lambda x: x['id'], results)
[u'1', u'2', u'3', u'4']
```

The boolean query also allows you to create more complex queries. You can OR term together using “|”, AND terms together using “+” or a space, and you can remove words from the query using the “-” operator:

```
>>> results = search_service.search(bq="'watched|moved'")
>>> results.hits
2
>>> map(lambda x: x['id'], results)
[u'3', u'4']
```

By default, the search will return 10 terms but it is possible to adjust this by using the `size` argument as follows:

```
>>> results = search_service.search(bq="'dan*', size=2)
>>> results.hits
```

(continues on next page)

(continued from previous page)

```
4
>>> map(lambda x: x['id'], results)
[u'1', u'2']
```

It is also possible to offset the start of the search by using the start argument as follows:

```
>>> results = search_service.search(bq="'dan*', start=2)
>>> results.hits
4
>>> map(lambda x: x['id'], results)
[u'3', u'4']
```

4.91.7 Ordering search results and rank expressions

If your search query is going to return many results, it is good to be able to sort them. You can order your search results by using the rank argument. You are able to sort on any fields which have the results option turned on:

```
>>> results = search_service.search(bq=query, rank=['-follower_count'])
```

You can also create your own rank expressions to sort your results according to other criteria, such as showing most recently active user, or combining the recency score with the text_relevance:

```
>>> domain.create_rank_expression('recently_active', 'last_activity')
>>> domain.create_rank_expression('activish',
...   'text_relevance + ((follower_count/(time() - last_activity))*1000)')
>>> results = search_service.search(bq=query, rank=['-recently_active'])
```

4.91.8 Viewing and Adjusting Stemming for a Domain

A stemming dictionary maps related words to a common stem. A stem is typically the root or base word from which variants are derived. For example, run is the stem of running and ran. During indexing, Amazon CloudSearch uses the stemming dictionary when it performs text-processing on text fields. At search time, the stemming dictionary is used to perform text-processing on the search request. This enables matching on variants of a word. For example, if you map the term running to the stem run and then search for running, the request matches documents that contain run as well as running.

To get the current stemming dictionary defined for a domain, use the `get_stemming` method:

```
>>> stems = domain.get_stemming()
>>> stems
{u'stems': {}}
```

This returns a dictionary object that can be manipulated directly to add additional stems for your search domain by adding pairs of term:stem to the stems dictionary:

```
>>> stems['stems']['running'] = 'run'
>>> stems['stems']['ran'] = 'run'
>>> stems
{u'stems': {u'ran': u'run', u'running': u'run'}}
>>>
```

This has changed the value locally. To update the information in Amazon CloudSearch, you need to save the data:

```
>>> stems.save()
```

You can also access certain CloudSearch-specific attributes related to the stemming dictionary defined for your domain:

```
>>> stems.status
u'RequiresIndexDocuments'
>>> stems.creation_date
u'2012-05-01T12:12:32Z'
>>> stems.update_date
u'2012-05-01T12:12:32Z'
>>> stems.update_version
19
>>>
```

The status indicates that, because you have changed the stems associated with the domain, you will need to re-index the documents in the domain before the new stems are used.

4.91.9 Viewing and Adjusting Stopwords for a Domain

Stopwords are words that should typically be ignored both during indexing and at search time because they are either insignificant or so common that including them would result in a massive number of matches.

To view the stopwords currently defined for your domain, use the `get_stopwords` method:

```
>>> stopwords = domain.get_stopwords()
>>> stopwords
{'stopwords': [u'a',
u'an',
u'and',
u'are',
u'as',
u'at',
u'be',
u'but',
u'by',
u'for',
u'in',
u'is',
u'it',
u'of',
u'on',
u'or',
u'the',
u'to',
u'was']}
```

You can add additional stopwords by simply appending the values to the list:

```
>>> stopwords['stopwords'].append('foo')
>>> stopwords['stopwords'].append('bar')
>>> stopwords
```

Similarly, you could remove currently defined stopwords from the list. To save the changes, use the `save` method:

```
>>> stopwords.save()
```

The stopwords object has similar attributes defined above for stemming that provide additional information about the stopwords in your domain.

4.91.10 Viewing and Adjusting Synonyms for a Domain

You can configure synonyms for terms that appear in the data you are searching. That way, if a user searches for the synonym rather than the indexed term, the results will include documents that contain the indexed term.

If you want two terms to match the same documents, you must define them as synonyms of each other. For example:

```
cat, feline
feline, cat
```

To view the synonyms currently defined for your domain, use the `get_synonyms` method:

```
>>> synonyms = domain.get_synonyms()
>>> synonyms
{u'synonyms': {}}
```

You can define new synonyms by adding new term:synonyms entries to the synonyms dictionary object:

```
>>> synonyms['synonyms']['cat'] = ['feline', 'kitten']
>>> synonyms['synonyms']['dog'] = ['canine', 'puppy']
```

To save the changes, use the `save` method:

```
>>> synonyms.save()
```

The synonyms object has similar attributes defined above for stemming that provide additional information about the stopwords in your domain.

4.91.11 Deleting Documents

It is also possible to delete documents:

```
>>> import time
>>> from datetime import datetime

>>> doc_service = domain.get_document_service()

>>> # Again we'll cheat and use the current epoch time as our version number

>>> doc_service.delete(4, int(time.mktime(datetime.utcnow().timetuple())))
>>> doc_service.commit()
```

4.92 CloudWatch

First, make sure you have something to monitor. You can either create a LoadBalancer or enable monitoring on an existing EC2 instance. To enable monitoring, you can either call the `monitor_instance` method on the `EC2Connection` object or call the `monitor` method on the `Instance` object.

It takes a while for the monitoring data to start accumulating but once it does, you can do this:

```
>>> import boto.ec2.cloudwatch
>>> c = boto.ec2.cloudwatch.connect_to_region('us-west-2')
>>> metrics = c.list_metrics()
>>> metrics
[Metric:DiskReadBytes,
 Metric:CPUUtilization,
 Metric:DiskWriteOps,
 Metric:DiskWriteOps,
 Metric:DiskReadOps,
 Metric:DiskReadBytes,
 Metric:DiskReadOps,
 Metric:CPUUtilization,
 Metric:DiskWriteOps,
 Metric:NetworkIn,
 Metric:NetworkOut,
 Metric:NetworkIn,
 Metric:DiskReadBytes,
 Metric:DiskWriteBytes,
 Metric:DiskWriteBytes,
 Metric:NetworkIn,
 Metric:NetworkIn,
 Metric:NetworkOut,
 Metric:NetworkOut,
 Metric:DiskReadOps,
 Metric:CPUUtilization,
 Metric:DiskReadOps,
 Metric:CPUUtilization,
 Metric:DiskWriteBytes,
 Metric:DiskWriteBytes,
 Metric:DiskReadBytes,
 Metric:NetworkOut,
 Metric:DiskWriteOps]
```

The `list_metrics` call will return a list of all of the available metrics that you can query against. Each entry in the list is a `Metric` object. As you can see from the list above, some of the metrics are repeated. The repeated metrics are across different dimensions (per-instance, per-image type, per instance type) which can be identified by looking at the `dimensions` property.

Because for this example, I'm only monitoring a single instance, the set of metrics available to me are fairly limited. If I was monitoring many instances, using many different instance types and AMI's and also several load balancers, the list of available metrics would grow considerably.

Once you have the list of available metrics, you can actually query the CloudWatch system for that metric. Let's choose the CPU utilization metric for one of the ImageID's:

```
>>> m_image = metrics[7]
>>> m_image
Metric:CPUUtilization
>>> m_image.dimensions
{'ImageId': [u'ami-6ac2a85a']}
```

Let's choose another CPU utilization metric for our instance.:

```
>>> m = metrics[20]
>>> m
Metric:CPUUtilization
```

(continues on next page)

(continued from previous page)

```
>>> m.dimensions
{'InstanceId': ['i-4ca81747']}
```

The Metric object has a query method that lets us actually perform the query against the collected data in CloudWatch. To call that, we need a start time and end time to control the time span of data that we are interested in. For this example, let's say we want the data for the previous hour:

```
>>> import datetime
>>> end = datetime.datetime.utcnow()
>>> start = end - datetime.timedelta(hours=1)
```

We also need to supply the Statistic that we want reported and the Units to use for the results. The Statistic can be one of these values:

```
['Minimum', 'Maximum', 'Sum', 'Average', 'SampleCount']
```

And Units must be one of the following:

```
['Seconds', 'Microseconds', 'Milliseconds', 'Bytes', 'Kilobytes', 'Megabytes',
↪ 'Gigabytes', 'Terabytes', 'Bits', 'Kilobits', 'Megabits', 'Gigabits', 'Terabits',
↪ 'Percent', 'Count', 'Bytes/Second', 'Kilobytes/Second', 'Megabytes/Second',
↪ 'Gigabytes/Second', 'Terabytes/Second', 'Bits/Second', 'Kilobits/Second', 'Megabits/
↪ Second', 'Gigabits/Second', 'Terabits/Second', 'Count/Second', None]
```

The query method also takes an optional parameter, period. This parameter controls the granularity (in seconds) of the data returned. The smallest period is 60 seconds and the value must be a multiple of 60 seconds. So, let's ask for the average as a percent:

```
>>> datapoints = m.query(start, end, 'Average', 'Percent')
>>> len(datapoints)
60
```

Our period was 60 seconds and our duration was one hour so we should get 60 data points back and we can see that we did. Each element in the datapoints list is a DataPoint object which is a simple subclass of a Python dict object. Each Datapoint object contains all of the information available about that particular data point.:

```
>>> d = datapoints[0]
>>> d
{'Timestamp': datetime.datetime(2014, 6, 23, 22, 25),
 u'Average': 20.0,
 u'Unit': u'Percent'}
```

My server obviously isn't very busy right now!

4.93 An Introduction to boto's VPC interface

This tutorial is based on the examples in the Amazon Virtual Private Cloud Getting Started Guide (<http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/>). In each example, it tries to show the boto request that correspond to the AWS command line tools.

4.93.1 Creating a VPC connection

First, we need to create a new VPC connection:

```
>>> from boto.vpc import VPCCConnection
>>> c = VPCCConnection()
```

4.93.2 To create a VPC

Now that we have a VPC connection, we can create our first VPC.

```
>>> vpc = c.create_vpc('10.0.0.0/24')
>>> vpc
VPC:vpc-6b1fe402
>>> vpc.id
u'vpc-6b1fe402'
>>> vpc.state
u'pending'
>>> vpc.cidr_block
u'10.0.0.0/24'
>>> vpc.dhcp_options_id
u'default'
>>>
```

4.93.3 To create a subnet

The next step is to create a subnet to associate with your VPC.

```
>>> subnet = c.create_subnet(vpc.id, '10.0.0.0/25')
>>> subnet.id
u'subnet-6a1fe403'
>>> subnet.state
u'pending'
>>> subnet.cidr_block
u'10.0.0.0/25'
>>> subnet.available_ip_address_count
123
>>> subnet.availability_zone
u'us-east-1b'
>>>
```

4.93.4 To create a customer gateway

Next, we create a customer gateway.

```
>>> cg = c.create_customer_gateway('ipsec.1', '12.1.2.3', 65534)
>>> cg.id
u'cgw-b6a247df'
>>> cg.type
u'ipsec.1'
>>> cg.state
u'available'
>>> cg.ip_address
u'12.1.2.3'
>>> cg.bgp_asn
u'65534'
>>>
```

4.93.5 To create a VPN gateway

```
>>> vg = c.create_vpn_gateway('ipsec.1')
>>> vg.id
u'vgw-44ad482d'
>>> vg.type
u'ipsec.1'
>>> vg.state
u'pending'
>>> vg.availability_zone
u'us-east-1b'
>>>
```

4.93.6 Attaching a VPN Gateway to a VPC

```
>>> vg.attach(vpc.id)
>>>
```

4.93.7 Associating an Elastic IP with a VPC Instance

```
>>> ec2.connection.associate_address('i-71b2f60b', None, 'eipalloc-35cf685d')
>>>
```

4.93.8 Releasing an Elastic IP Attached to a VPC Instance

```
>>> ec2.connection.release_address(None, 'eipalloc-35cf685d')
>>>
```

4.93.9 To Get All VPN Connections

```
>>> vpns = c.get_all_vpn_connections()
>>> vpns[0].id
u'vpn-12ef67bv'
>>> tunnels = vpns[0].tunnels
>>> tunnels
[VpnTunnel: 177.12.34.56, VpnTunnel: 177.12.34.57]
```

4.93.10 To Create VPC Peering Connection

```
>>> vpcs = c.get_all_vpcs()
>>> vpc_peering_connection = c.create_vpc_peering_connection(vpcs[0].id, vpcs[1].id)
>>> vpc_peering_connection
VpcPeeringConnection:pcx-18987471
```

4.93.11 To Accept VPC Peering Connection

```
>>> vpc_peering_connections = c.get_all_vpc_peering_connections()
>>> vpc_peering_connection = vpc_peering_connections[0]
>>> vpc_peering_connection.status_code
u'pending-acceptance'
>>> vpc_peering_connection = c.accept_vpc_peering_connection(vpc_peering_connection.
↳ id)
>>> vpc_peering_connection.update()
u'active'
```

4.93.12 To Reject VPC Peering Connection

```
>>> vpc_peering_connections = c.get_all_vpc_peering_connections()
>>> vpc_peering_connection = vpc_peering_connections[0]
>>> vpc_peering_connection.status_code
u'pending-acceptance'
>>> c.reject_vpc_peering_connection(vpc_peering_connection.id)
>>> vpc_peering_connection.update()
u'rejected'
```

4.94 An Introduction to boto's Elastic Load Balancing interface

This tutorial focuses on the boto interface for [Elastic Load Balancing](#) from Amazon Web Services. This tutorial assumes that you have already downloaded and installed boto, and are familiar with the boto ec2 interface.

4.94.1 Elastic Load Balancing Concepts

[Elastic Load Balancing](#) (ELB) is intimately connected with Amazon's [Elastic Compute Cloud](#) (EC2) service. Using the ELB service allows you to create a load balancer - a DNS endpoint and set of ports that distributes incoming requests to a set of EC2 instances. The advantages of using a load balancer is that it allows you to truly scale up or down a set of backend instances without disrupting service. Before the ELB service, you had to do this manually by launching an EC2 instance and installing load balancer software on it (nginx, haproxy, perlbal, etc.) to distribute traffic to other EC2 instances.

Recall that the EC2 service is split into Regions, which are further divided into Availability Zones (AZ). For example, the US-East region is divided into us-east-1a, us-east-1b, us-east-1c, us-east-1d, and us-east-1e. You can think of AZs as data centers - each runs off a different set of ISP backbones and power providers. ELB load balancers can span multiple AZs but cannot span multiple regions. That means that if you'd like to create a set of instances spanning both the US and Europe Regions you'd have to create two load balancers and have some sort of other means of distributing requests between the two load balancers. An example of this could be using GeoIP techniques to choose the correct load balancer, or perhaps DNS round robin. Keep in mind also that traffic is distributed equally over all AZs the ELB balancer spans. This means you should have an equal number of instances in each AZ if you want to equally distribute load amongst all your instances.

4.94.2 Creating a Connection

The first step in accessing ELB is to create a connection to the service.

Like EC2, the ELB service has a different endpoint for each region. By default the US East endpoint is used. To choose a specific region, use the `connect_to_region` function:

```
>>> import boto.ec2.elb
>>> elb = boto.ec2.elb.connect_to_region('us-west-2')
```

Here's yet another way to discover what regions are available and then connect to one:

```
>>> import boto.ec2.elb
>>> regions = boto.ec2.elb.regions()
>>> regions
[RegionInfo:us-east-1,
 RegionInfo:ap-northeast-1,
 RegionInfo:us-west-1,
 RegionInfo:us-west-2,
 RegionInfo:ap-southeast-1,
 RegionInfo:eu-west-1]
>>> elb = regions[-1].connect()
```

Alternatively, edit your `boto.cfg` with the default ELB endpoint to use:

```
[Boto]
elb_region_name = eu-west-1
elb_region_endpoint = elasticloadbalancing.eu-west-1.amazonaws.com
```

Getting Existing Load Balancers

To retrieve any existing load balancers:

```
>>> conn.get_all_load_balancers()
[LoadBalancer:load-balancer-prod, LoadBalancer:load-balancer-staging]
```

You can also filter by name

```
>>> conn.get_all_load_balancers(load_balancer_names=['load-balancer-prod'])
[LoadBalancer:load-balancer-prod]
```

`get_all_load_balancers` returns a `boto.resultset.ResultSet` that contains instances of `boto.ec2.elb.loadbalancer.LoadBalancer`, each of which abstracts access to a load balancer. `ResultSet` works very much like a list.

```
>>> balancers = conn.get_all_load_balancers()
>>> balancers[0]
LoadBalancer:load-balancer-prod
```

4.94.3 Creating a Load Balancer

To create a load balancer you need the following:

1. The specific **ports and protocols** you want to load balancer over, and what port you want to connect to all instances.
2. A **health check** - the ELB concept of a *heart beat* or *ping*. ELB will use this health check to see whether your instances are up or down. If they go down, the load balancer will no longer send requests to them.
3. A **list of Availability Zones** you'd like to create your load balancer over.

Ports and Protocols

An incoming connection to your load balancer will come on one or more ports - for example 80 (HTTP) and 443 (HTTPS). Each can be using a protocol - currently, the supported protocols are TCP and HTTP. We also need to tell the load balancer which port to route connects *to* on each instance. For example, to create a load balancer for a website that accepts connections on 80 and 443, and that routes connections to port 8080 and 8443 on each instance, you would specify that the load balancer ports and protocols are:

- 80, 8080, HTTP
- 443, 8443, TCP

This says that the load balancer will listen on two ports - 80 and 443. Connections on 80 will use an HTTP load balancer to forward connections to port 8080 on instances. Likewise, the load balancer will listen on 443 to forward connections to 8443 on each instance using the TCP balancer. We need to use TCP for the HTTPS port because it is encrypted at the application layer. Of course, we could specify the load balancer use TCP for port 80, however specifying HTTP allows you to let ELB handle some work for you - for example HTTP header parsing.

Configuring a Health Check

A health check allows ELB to determine which instances are alive and able to respond to requests. A health check is essentially a tuple consisting of:

- *Target*: What to check on an instance. For a TCP check this is comprised of:

```
TCP:PORT_TO_CHECK
```

Which attempts to open a connection on `PORT_TO_CHECK`. If the connection opens successfully, that specific instance is deemed healthy, otherwise it is marked temporarily as unhealthy. For HTTP, the situation is slightly different:

```
HTTP:PORT_TO_CHECK/RESOURCE
```

This means that the health check will connect to the resource `/RESOURCE` on `PORT_TO_CHECK`. If an HTTP 200 status is returned the instance is deemed healthy.

- *Interval*: How often the check is made. This is given in seconds and defaults to 30. The valid range of intervals goes from 5 seconds to 600 seconds.
- *Timeout*: The number of seconds the load balancer will wait for a check to return a result.
- *Unhealthy threshold*: The number of consecutive failed checks to deem the instance as being dead. The default is 5, and the range of valid values lies from 2 to 10.

The following example creates a health check called `instance_health` that simply checks instances every 20 seconds on port 80 over HTTP at the resource `/health` for 200 successes.

```
>>> from boto.ec2.elb import HealthCheck
>>> hc = HealthCheck(
    interval=20,
    healthy_threshold=3,
    unhealthy_threshold=5,
    target='HTTP:8080/health'
)
```

Putting It All Together

Finally, let's create a load balancer in the US region that listens on ports 80 and 443 and distributes requests to instances on 8080 and 8443 over HTTP and TCP. We want the load balancer to span the availability zones *us-east-1a* and *us-east-1b*:

```
>>> zones = ['us-east-1a', 'us-east-1b']
>>> ports = [(80, 8080, 'http'), (443, 8443, 'tcp')]
>>> lb = conn.create_load_balancer('my-lb', zones, ports)
>>> # This is from the previous section.
>>> lb.configure_health_check(hc)
```

The load balancer has been created. To see where you can actually connect to it, do:

```
>>> print lb.dns_name
my_elb-123456789.us-east-1.elb.amazonaws.com
```

You can then CNAME map a better name, i.e. www.MYWEBSITE.com to the above address.

4.94.4 Adding Instances To a Load Balancer

Now that the load balancer has been created, there are two ways to add instances to it:

1. Manually, adding each instance in turn.
2. Mapping an autoscale group to the load balancer. Please see the [Autoscale tutorial](#) for information on how to do this.

Manually Adding and Removing Instances

Assuming you have a list of instance ids, you can add them to the load balancer

```
>>> instance_ids = ['i-4f8cf126', 'i-0bb7ca62']
>>> lb.register_instances(instance_ids)
```

Keep in mind that these instances should be in Security Groups that match the internal ports of the load balancer you just created (for this example, they should allow incoming connections on 8080 and 8443).

To remove instances:

```
>>> lb.deregister_instances(instance_ids)
```

4.94.5 Modifying Availability Zones for a Load Balancer

If you wanted to disable one or more zones from an existing load balancer:

```
>>> lb.disable_zones(['us-east-1a'])
```

You can then terminate each instance in the disabled zone and then deregister them from your load balancer.

To enable zones:

```
>>> lb.enable_zones(['us-east-1c'])
```


4.94.6 Deleting a Load Balancer

```
>>> lb.delete()
```

4.95 An Introduction to boto's S3 interface

This tutorial focuses on the boto interface to the Simple Storage Service from Amazon Web Services. This tutorial assumes that you have already downloaded and installed boto.

4.95.1 Creating a Connection

The first step in accessing S3 is to create a connection to the service. There are two ways to do this in boto. The first is:

```
>>> from boto.s3.connection import S3Connection
>>> conn = S3Connection('<aws access key>', '<aws secret key>')
```

At this point the variable `conn` will point to an `S3Connection` object. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

- `AWS_ACCESS_KEY_ID` - Your AWS Access Key ID
- `AWS_SECRET_ACCESS_KEY` - Your AWS Secret Access Key

and then call the constructor without any arguments, like this:

```
>>> conn = S3Connection()
```

There is also a shortcut function in the boto package, called `connect_s3` that may provide a slightly easier means of creating a connection:

```
>>> import boto
>>> conn = boto.connect_s3()
```

In either case, `conn` will point to an `S3Connection` object which we will use throughout the remainder of this tutorial.

4.95.2 Creating a Bucket

Once you have a connection established with S3, you will probably want to create a bucket. A bucket is a container used to store key/value pairs in S3. A bucket can hold an unlimited amount of data so you could potentially have just one bucket in S3 for all of your information. Or, you could create separate buckets for different types of data. You can figure all of that out later, first let's just create a bucket. That can be accomplished like this:

```
>>> bucket = conn.create_bucket('mybucket')
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
  File "boto/connection.py", line 285, in create_bucket
    raise S3CreateError(response.status, response.reason)
boto.exception.S3CreateError: S3Error[409]: Conflict
```

Whoa. What happened there? Well, the thing you have to know about buckets is that they are kind of like domain names. It's one flat name space that everyone who uses S3 shares. So, someone has already create a bucket called "mybucket" in S3 and that means no one else can grab that bucket name. So, you have to come up with a name that

hasn't been taken yet. For example, something that uses a unique string as a prefix. Your `AWS_ACCESS_KEY` (NOT YOUR SECRET KEY!) could work but I'll leave it to your imagination to come up with something. I'll just assume that you found an acceptable name.

The `create_bucket` method will create the requested bucket if it does not exist or will return the existing bucket if it does exist.

4.95.3 Creating a Bucket In Another Location

The example above assumes that you want to create a bucket in the standard US region. However, it is possible to create buckets in other locations. To do so, first import the `Location` object from the `boto.s3.connection` module, like this:

```
>>> from boto.s3.connection import Location
>>> print '\n'.join(i for i in dir(Location) if i[0].isupper())
APNortheast
APSoutheast
APSoutheast2
DEFAULT
EU
EUCentral1
SAEast
USWest
USWest2
```

As you can see, the `Location` object defines a number of possible locations. By default, the location is the empty string which is interpreted as the US Classic Region, the original S3 region. However, by specifying another location at the time the bucket is created, you can instruct S3 to create the bucket in that location. For example:

```
>>> conn.create_bucket('mybucket', location=Location.EU)
```

will create the bucket in the EU region (assuming the name is available).

4.95.4 Storing Data

Once you have a bucket, presumably you will want to store some data in it. S3 doesn't care what kind of information you store in your objects or what format you use to store it. All you need is a key that is unique within your bucket.

The `Key` object is used in boto to keep track of data stored in S3. To store new data in S3, start by creating a new `Key` object:

```
>>> from boto.s3.key import Key
>>> k = Key(bucket)
>>> k.key = 'foobar'
>>> k.set_contents_from_string('This is a test of S3')
```

The net effect of these statements is to create a new object in S3 with a key of "foobar" and a value of "This is a test of S3". To validate that this worked, quit out of the interpreter and start it up again. Then:

```
>>> import boto
>>> c = boto.connect_s3()
>>> b = c.get_bucket('mybucket') # substitute your bucket name here
>>> from boto.s3.key import Key
>>> k = Key(b)
>>> k.key = 'foobar'
```

(continues on next page)

(continued from previous page)

```
>>> k.get_contents_as_string()
'This is a test of S3'
```

So, we can definitely store and retrieve strings. A more interesting example may be to store the contents of a local file in S3 and then retrieve the contents to another local file.

```
>>> k = Key(b)
>>> k.key = 'myfile'
>>> k.set_contents_from_filename('foo.jpg')
>>> k.get_contents_to_filename('bar.jpg')
```

There are a couple of things to note about this. When you send data to S3 from a file or filename, boto will attempt to determine the correct mime type for that file and send it as a Content-Type header. The boto package uses the standard mimetypes package in Python to do the mime type guessing. The other thing to note is that boto does stream the content to and from S3 so you should be able to send and receive large files without any problem.

When fetching a key that already exists, you have two options. If you're uncertain whether a key exists (or if you need the metadata set on it, you can call `Bucket.get_key(key_name_here)`. However, if you're sure a key already exists within a bucket, you can skip the check for a key on the server.

```
>>> import boto
>>> c = boto.connect_s3()
>>> b = c.get_bucket('mybucket') # substitute your bucket name here

# Will hit the API to check if it exists.
>>> possible_key = b.get_key('mykey') # substitute your key name here

# Won't hit the API.
>>> key_we_know_is_there = b.get_key('mykey', validate=False)
```

4.95.5 Storing Large Data

At times the data you may want to store will be hundreds of megabytes or more in size. S3 allows you to split such files into smaller components. You upload each component in turn and then S3 combines them into the final object. While this is fairly straightforward, it requires a few extra steps to be taken. The example below makes use of the FileChunkIO module, so pip install FileChunkIO if it isn't already installed.

```
>>> import math, os
>>> import boto
>>> from filechunkio import FileChunkIO

# Connect to S3
>>> c = boto.connect_s3()
>>> b = c.get_bucket('mybucket')

# Get file info
>>> source_path = 'path/to/your/file.ext'
>>> source_size = os.stat(source_path).st_size

# Create a multipart upload request
>>> mp = b.initiate_multipart_upload(os.path.basename(source_path))

# Use a chunk size of 50 MiB (feel free to change this)
>>> chunk_size = 52428800
```

(continues on next page)

(continued from previous page)

```

>>> chunk_count = int(math.ceil(source_size / float(chunk_size)))

# Send the file parts, using FileChunkIO to create a file-like object
# that points to a certain byte range within the original file. We
# set bytes to never exceed the original file size.
>>> for i in range(chunk_count):
>>>     offset = chunk_size * i
>>>     bytes = min(chunk_size, source_size - offset)
>>>     with FileChunkIO(source_path, 'r', offset=offset,
>>>                     bytes=bytes) as fp:
>>>         mp.upload_part_from_file(fp, part_num=i + 1)

# Finish the upload
>>> mp.complete_upload()

```

It is also possible to upload the parts in parallel using threads. The `s3put` script that ships with Boto provides an example of doing so using a thread pool.

Note that if you forget to call either `mp.complete_upload()` or `mp.cancel_upload()` you will be left with an incomplete upload and charged for the storage consumed by the uploaded parts. A call to `bucket.get_all_multipart_uploads()` can help to show lost multipart upload parts.

4.95.6 Accessing A Bucket

Once a bucket exists, you can access it by getting the bucket. For example:

```

>>> mybucket = conn.get_bucket('mybucket') # Substitute in your bucket name
>>> mybucket.list()
...listing of keys in the bucket...

```

By default, this method tries to validate the bucket's existence. You can override this behavior by passing `validate=False`:

```

>>> nonexistent = conn.get_bucket('i-dont-exist-at-all', validate=False)

```

Changed in version 2.25.0.

Warning: If `validate=False` is passed, no request is made to the service (no charge/communication delay). This is only safe to do if you are **sure** the bucket exists.

If the default `validate=True` is passed, a request is made to the service to ensure the bucket exists. Prior to Boto v2.25.0, this fetched a list of keys (but with a max limit set to 0, always returning an empty list) in the bucket (& included better error messages), at an increased expense. As of Boto v2.25.0, this now performs a HEAD request (less expensive but worse error messages).

If you were relying on parsing the error message before, you should call something like:

```

bucket = conn.get_bucket('<bucket_name>', validate=False)
bucket.get_all_keys(maxkeys=0)

```

If the bucket does not exist, a `S3ResponseError` will commonly be thrown. If you'd rather not deal with any exceptions, you can use the `lookup` method.:

```
>>> nonexistent = conn.lookup('i-dont-exist-at-all')
>>> if nonexistent is None:
...     print "No such bucket!"
...
No such bucket!
```

4.95.7 Deleting A Bucket

Removing a bucket can be done using the `delete_bucket` method. For example:

```
>>> conn.delete_bucket('mybucket') # Substitute in your bucket name
```

The bucket must be empty of keys or this call will fail & an exception will be raised. You can remove a non-empty bucket by doing something like:

```
>>> full_bucket = conn.get_bucket('bucket-to-delete')
# It's full of keys. Delete them all.
>>> for key in full_bucket.list():
...     key.delete()
...
# The bucket is empty now. Delete it.
>>> conn.delete_bucket('bucket-to-delete')
```

Warning: This method can cause data loss! Be very careful when using it.

Additionally, be aware that using the above method for removing all keys and deleting the bucket involves a request for each key. As such, it's not particularly fast & is very chatty.

4.95.8 Listing All Available Buckets

In addition to accessing specific buckets via the `create_bucket` method you can also get a list of all available buckets that you have created.

```
>>> rs = conn.get_all_buckets()
```

This returns a `ResultSet` object (see the [SQS Tutorial](#) for more info on `ResultSet` objects). The `ResultSet` can be used as a sequence or list type object to retrieve `Bucket` objects.

```
>>> len(rs)
11
>>> for b in rs:
...     print b.name
...
<listing of available buckets>
>>> b = rs[0]
```

4.95.9 Setting / Getting the Access Control List for Buckets and Keys

The S3 service provides the ability to control access to buckets and keys within s3 via the Access Control List (ACL) associated with each object in S3. There are two ways to set the ACL for an object:

1. Create a custom ACL that grants specific rights to specific users. At the moment, the users that are specified within grants have to be registered users of Amazon Web Services so this isn't as useful or as general as it could be.
2. Use a "canned" access control policy. There are four canned policies defined:
 - (a) private: Owner gets FULL_CONTROL. No one else has any access rights.
 - (b) public-read: Owners gets FULL_CONTROL and the anonymous principal is granted READ access.
 - (c) public-read-write: Owner gets FULL_CONTROL and the anonymous principal is granted READ and WRITE access.
 - (d) authenticated-read: Owner gets FULL_CONTROL and any principal authenticated as a registered Amazon S3 user is granted READ access.

To set a canned ACL for a bucket, use the `set_acl` method of the Bucket object. The argument passed to this method must be one of the four permissible canned policies named in the list `CannedACLStrings` contained in `acl.py`. For example, to make a bucket readable by anyone:

```
>>> b.set_acl('public-read')
```

You can also set the ACL for Key objects, either by passing an additional argument to the above method:

```
>>> b.set_acl('public-read', 'foobar')
```

where 'foobar' is the key of some object within the bucket `b` or you can call the `set_acl` method of the Key object:

```
>>> k.set_acl('public-read')
```

You can also retrieve the current ACL for a Bucket or Key object using the `get_acl` object. This method parses the `AccessControlPolicy` response sent by S3 and creates a set of Python objects that represent the ACL.

```
>>> acp = b.get_acl()
>>> acp
<boto.acl.Policy instance at 0x2e6940>
>>> acp.acl
<boto.acl.ACL instance at 0x2e69e0>
>>> acp.acl.grants
[<boto.acl.Grant instance at 0x2e6a08>]
>>> for grant in acp.acl.grants:
...     print grant.permission, grant.display_name, grant.email_address, grant.id
...
FULL_CONTROL <boto.user.User instance at 0x2e6a30>
```

The Python objects representing the ACL can be found in the `acl.py` module of boto.

Both the Bucket object and the Key object also provide shortcut methods to simplify the process of granting individuals specific access. For example, if you want to grant an individual user READ access to a particular object in S3 you could do the following:

```
>>> key = b.lookup('mykeytosshare')
>>> key.add_email_grant('READ', 'foo@bar.com')
```

The email address provided should be the one associated with the users AWS account. There is a similar method called `add_user_grant` that accepts the canonical id of the user rather than the email address.

4.95.10 Setting/Getting Metadata Values on Key Objects

S3 allows arbitrary user metadata to be assigned to objects within a bucket. To take advantage of this S3 feature, you should use the `set_metadata` and `get_metadata` methods of the Key object to set and retrieve metadata associated with an S3 object. For example:

```
>>> k = Key(b)
>>> k.key = 'has_metadata'
>>> k.set_metadata('meta1', 'This is the first metadata value')
>>> k.set_metadata('meta2', 'This is the second metadata value')
>>> k.set_contents_from_filename('foo.txt')
```

This code associates two metadata key/value pairs with the Key `k`. To retrieve those values later:

```
>>> k = b.get_key('has_metadata')
>>> k.get_metadata('meta1')
'This is the first metadata value'
>>> k.get_metadata('meta2')
'This is the second metadata value'
>>>
```

4.95.11 Setting/Getting/Deleting CORS Configuration on a Bucket

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support in Amazon S3, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

To create a CORS configuration and associate it with a bucket:

```
>>> from boto.s3.cors import CORSConfiguration
>>> cors_cfg = CORSConfiguration()
>>> cors_cfg.add_rule(['PUT', 'POST', 'DELETE'], 'https://www.example.com', allowed_
↳header='*', max_age_seconds=3000, expose_header='x-amz-server-side-encryption')
>>> cors_cfg.add_rule('GET', '*')
```

The above code creates a CORS configuration object with two rules.

- The first rule allows cross-origin PUT, POST, and DELETE requests from the `https://www.example.com/` origin. The rule also allows all headers in preflight OPTIONS request through the Access-Control-Request-Headers header. In response to any preflight OPTIONS request, Amazon S3 will return any requested headers.
- The second rule allows cross-origin GET requests from all origins.

To associate this configuration with a bucket:

```
>>> import boto
>>> c = boto.connect_s3()
>>> bucket = c.lookup('mybucket')
>>> bucket.set_cors(cors_cfg)
```

To retrieve the CORS configuration associated with a bucket:

```
>>> cors_cfg = bucket.get_cors()
```

And, finally, to delete all CORS configurations from a bucket:

```
>>> bucket.delete_cors()
```

4.95.12 Transitioning Objects

S3 buckets support transitioning objects to various storage classes. This is done using lifecycle policies. You can currently transition objects to Infrequent Access, Glacier, or just plain Expire. All of these options are capable of being applied after a number of days or after a given date. Lifecycle configurations are assigned to buckets and require these parameters:

- The object prefix that identifies the objects you are targeting. (or none)
- The action you want S3 to perform on the identified objects.
- The date or number of days when you want S3 to perform these actions.

For example, given a bucket `s3-lifecycle-boto-demo`, we can first retrieve the bucket:

```
>>> import boto
>>> c = boto.connect_s3()
>>> bucket = c.get_bucket('s3-lifecycle-boto-demo')
```

Then we can create a lifecycle object. In our example, we want all objects under `logs/*` to transition to Standard IA 30 days after the object is created, glacier 90 days after creation, and be deleted 120 days after creation.

```
>>> from boto.s3.lifecycle import Lifecycle, Transitions, Rule
>>> transitions = Transitions()
>>> transitions.add_transition(days=30, storage_class='STANDARD_IA')
>>> transitions.add_transition(days=90, storage_class='GLACIER')
>>> expiration = Expiration(days=120)
>>> rule = Rule(id='ruleid', prefix='logs/', status='Enabled', expiration=expiration,
↳ transition=transitions)
>>> lifecycle = Lifecycle()
>>> lifecycle.append(rule)
```

Note: For API docs for the lifecycle objects, see [*boto.s3.lifecycle*](#)

We can now configure the bucket with this lifecycle policy:

```
>>> bucket.configure_lifecycle(lifecycle)
True
```

You can also retrieve the current lifecycle policy for the bucket:

```
>>> current = bucket.get_lifecycle_config()
>>> print current[0].transition
>>> print current[0].expiration
[<Transition: in: 90 days, GLACIER>, <Transition: in: 30 days, STANDARD_IA>]
<Expiration: in: 120 days>
```

Note: We have deprecated directly accessing transition properties from the lifecycle object. You must index into the transition array first.

When an object transitions, the storage class will be updated. This can be seen when you **list** the objects in a bucket:


```
>>> for key in bucket.list():
...     print key, key.storage_class
...
<Key: s3-lifecycle-boto-demo,logs/testlog1.log> STANDARD_IA
<Key: s3-lifecycle-boto-demo,logs/testlog2.log> GLACIER
```

You can also use the prefix argument to the `bucket.list` method:

```
>>> print list(bucket.list(prefix='logs/testlog1.log'))[0].storage_class
>>> print list(bucket.list(prefix='logs/testlog2.log'))[0].storage_class
u'STANDARD_IA'
u'GLACIER'
```

4.95.13 Restoring Objects from Glacier

Once an object has been transitioned to Glacier, you can restore the object back to S3. To do so, you can use the `boto.s3.key.Key.restore()` method of the key object. The `restore` method takes an integer that specifies the number of days to keep the object in S3.

```
>>> import boto
>>> c = boto.connect_s3()
>>> bucket = c.get_bucket('s3-glacier-boto-demo')
>>> key = bucket.get_key('logs/testlog1.log')
>>> key.restore(days=5)
```

It takes about 4 hours for a restore operation to make a copy of the archive available for you to access. While the object is being restored, the `ongoing_restore` attribute will be set to `True`:

```
>>> key = bucket.get_key('logs/testlog1.log')
>>> print key.ongoing_restore
True
```

When the restore is finished, this value will be `False` and the expiry date of the object will be non `None`:

```
>>> key = bucket.get_key('logs/testlog1.log')
>>> print key.ongoing_restore
False
>>> print key.expiry_date
"Fri, 21 Dec 2012 00:00:00 GMT"
```

Note: If there is no restore operation either in progress or completed, the `ongoing_restore` attribute will be `None`.

Once the object is restored you can then download the contents:

```
>>> key.get_contents_to_filename('testlog1.log')
```

4.96 An Introduction to boto's Route53 interface

This tutorial focuses on the boto interface to Route53 from Amazon Web Services. This tutorial assumes that you have already downloaded and installed boto.

Route53 is a Domain Name System (DNS) web service. It can be used to route requests to services running on AWS such as EC2 instances or load balancers, as well as to external services. Route53 also allows you to have automated checks to send requests where you require them.

In this tutorial, we will be setting up our services for *example.com*.

4.96.1 Creating a connection

To start using Route53 you will need to create a connection to the service as normal:

```
>>> import boto.route53
>>> conn = boto.route53.connect_to_region('us-west-2')
```

You will be using this `conn` object for the remainder of the tutorial to send commands to Route53.

4.96.2 Working with domain names

You can manipulate domains through a zone object. For example, you can create a domain name:

```
>>> zone = conn.create_zone("example.com.")
```

Note that trailing dot on that domain name is significant. This is known as a fully qualified domain name (FQDN).

```
>>> zone
<Zone:example.com.>
```

You can also retrieve all your domain names:

```
>>> conn.get_zones()
[<Zone:example.com.>]
```

Or you can retrieve a single domain:

```
>>> conn.get_zone("example.com.")
<Zone:example.com.>
```

Finally, you can retrieve the list of nameservers that AWS has setup for this domain name as follows:

```
>>> zone.get_nameservers()
[u'ns-1000.awsdns-42.org.', u'ns-1001.awsdns-30.com.', u'ns-1002.awsdns-59.net.', u
↪ 'ns-1003.awsdns-09.co.uk.']
```

Once you have finished configuring your domain name, you will need to change your nameservers at your registrar to point to those nameservers for Route53 to work.

4.96.3 Setting up dumb records

You can also add, update and delete records on a zone:

```
>>> status = a.add_record("MX", "example.com.", "10 mail.isp.com")
```

When you send a change request through, the status of the update will be *PENDING*:

```
>>> status
<Status:PENDING>
```

You can call the API again and ask for the current status as follows:

```
>>> status.update()
'INSYNC'
```

```
>>> status
<Status:INSYNC>
```

When the status has changed to *INSYNC*, the change has been propagated to remote servers

4.96.4 Updating a record

You can create, upsert or delete a single record like this

```
>>> zone = conn.get_zone("example.com.")
>>> change_set = ResourceRecordSets(conn, zone.id)
>>> changes1 = change_set.add_change("UPSERT", "www" + ".example.com", type="CNAME",
↳ttl=3600)
>>> changes1.add_value("webserver.example.com")
>>> change_set.commit()
```

In this example we create or update, depending on the existence of the record, the CNAME `www.example.com` to `webserver.example.com`.

4.96.5 Working with Change Sets

You can also do bulk updates using `ResourceRecordSets`. For example updating the TTL

```
>>> zone = conn.get_zone('example.com')
>>> change_set = boto.route53.record.ResourceRecordSets(conn, zone.id)
>>> for rrset in conn.get_all_rrsets(zone.id):
...     u = change_set.add_change("UPSERT", rrset.name, rrset.type, ttl=3600)
...     u.add_value(rrset.resource_records[0])
... results = change_set.commit()
Done
```

In this example we update the TTL to 1hr (3600 seconds) for all records recursed from `example.com`. Note: this will also change the SOA and NS records which may not be ideal for many users.

4.97 Boto Config

4.97.1 Introduction

There is a growing list of configuration options for the boto library. Many of these options can be passed into the constructors for top-level objects such as connections. Some options, such as credentials, can also be read from environment variables (e.g. `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_SECURITY_TOKEN` and `AWS_PROFILE`). It is also possible to manage these options in a central place through the use of boto config files.

4.97.2 Details

A boto config file is a text file formatted like an .ini configuration file that specifies values for options that control the behavior of the boto library. In Unix/Linux systems, on startup, the boto library looks for configuration files in the following locations and in the following order:

- /etc/boto.cfg - for site-wide settings that all users on this machine will use
- (if profile is given) ~/.aws/credentials - for credentials shared between SDKs
- (if profile is given) ~/.boto - for user-specific settings
- ~/.aws/credentials - for credentials shared between SDKs
- ~/.boto - for user-specific settings

Comments You can comment out a line by putting a '#' at the beginning of the line, just like in Python code.

In Windows, create a text file that has any name (e.g. boto.config). It's recommended that you put this file in your user folder. Then set a user environment variable named BOTO_CONFIG to the full path of that file.

The options in the config file are merged into a single, in-memory configuration that is available as `boto.config`. The `boto.pyami.config.Config` class is a subclass of the standard Python `ConfigParser.SafeConfigParser` object and inherits all of the methods of that object. In addition, the boto `Config` class defines additional methods that are described on the [PyamiConfigMethods](#) page.

An example boto config file might look like:

```
[Credentials]
aws_access_key_id = <your_access_key_here>
aws_secret_access_key = <your_secret_key_here>
```

4.97.3 Sections

The following sections and options are currently recognized within the boto config file.

Credentials

The Credentials section is used to specify the AWS credentials used for all boto requests. The order of precedence for authentication credentials is:

- Credentials passed into the Connection class constructor.
- Credentials specified by environment variables
- Credentials specified as named profiles in the shared credential file.
- Credentials specified by default in the shared credential file.
- Credentials specified as named profiles in the config file.
- Credentials specified by default in the config file.

This section defines the following options: `aws_access_key_id` and `aws_secret_access_key`. The former being your AWS key id and the latter being the secret key.

For example:

```
[profile name_goes_here]
aws_access_key_id = <access key for this profile>
aws_secret_access_key = <secret key for this profile>

[Credentials]
aws_access_key_id = <your default access key>
aws_secret_access_key = <your default secret key>
```

Please notice that quote characters are not used to either side of the '=' operator even when both your AWS access key ID and secret key are strings.

If you have multiple AWS keypairs that you use for different purposes, use the `profile` style shown above. You can set an arbitrary number of profiles within your configuration files and then reference them by name when you instantiate your connection. If you specify a profile that does not exist in the configuration, the keys used under the `[Credentials]` heading will be applied by default.

The shared credentials file in `~/.aws/credentials` uses a slightly different format. For example:

```
[default]
aws_access_key_id = <your default access key>
aws_secret_access_key = <your default secret key>

[name_goes_here]
aws_access_key_id = <access key for this profile>
aws_secret_access_key = <secret key for this profile>

[another_profile]
aws_access_key_id = <access key for this profile>
aws_secret_access_key = <secret key for this profile>
aws_security_token = <optional security token for this profile>
```

For greater security, the secret key can be stored in a keyring and retrieved via the keyring package. To use a keyring, use `keyring`, rather than `aws_secret_access_key`:

```
[Credentials]
aws_access_key_id = <your access key>
keyring = <keyring name>
```

To use a keyring, you must have the Python `keyring` package installed and in the Python path. To learn about setting up keyrings, see the [keyring documentation](#)

Credentials can also be supplied for a Eucalyptus service:

```
[Credentials]
euca_access_key_id = <your access key>
euca_secret_access_key = <your secret key>
```

Finally, this section is also be used to provide credentials for the Internet Archive API:

```
[Credentials]
ia_access_key_id = <your access key>
ia_secret_access_key = <your secret key>
```

Boto

The Boto section is used to specify options that control the operation of boto itself. This section defines the following options:

debug Controls the level of debug messages that will be printed by the boto library. The following values are defined:

```
0 - no debug messages are printed
1 - basic debug messages from boto are printed
2 - all boto debugging messages plus request/response messages from
  ↳httplib
```

proxy The name of the proxy host to use for connecting to AWS.

proxy_port The port number to use to connect to the proxy host.

proxy_user The user name to use when authenticating with proxy host.

proxy_pass The password to use when authenticating with proxy host.

num_retries The number of times to retry failed requests to an AWS server. If boto receives an error from AWS, it will attempt to recover and retry the request. The default number of retries is 5 but you can change the default with this option.

For example:

```
[Boto]
debug = 0
num_retries = 10

proxy = myproxy.com
proxy_port = 8080
proxy_user = foo
proxy_pass = bar
```

connection_stale_duration Amount of time to wait in seconds before a connection will stop getting reused. AWS will disconnect connections which have been idle for 180 seconds.

is_secure Is the connection over SSL. This setting will override passed in values.

https_validate_certificates Validate HTTPS certificates. This is on by default

ca_certificates_file Location of CA certificates or the keyword “system”. Using the system keyword lets boto get out of the way and makes the SSL certificate validation the responsibility the underlying SSL implementation provided by the system.

http_socket_timeout Timeout used to overwrite the system default socket timeout for http lib .

send_crlf_after_proxy_auth_headers Change line ending behaviour with proxies. For more details see [this discussion](#)

endpoints_path Allows customizing the regions/endpoints available in Boto. Provide an absolute path to a custom JSON file, which gets merged into the defaults. (This can also be specified with the BOTO_ENDPOINTS environment variable instead.)

use_endpoint_heuristics Allows using endpoint heuristics to guess endpoints for regions that aren't built in. This can also be specified with the BOTO_USE_ENDPOINT_HEURISTICS environment variable.

These settings will default to:

```
[Boto]
connection_stale_duration = 180
is_secure = True
https_validate_certificates = True
ca_certificates_file = cacerts.txt
```

(continues on next page)

(continued from previous page)

```

http_socket_timeout = 60
send_crlf_after_proxy_auth_headers = False
endpoints_path = /path/to/my/boto/endpoints.json
use_endpoint_heuristics = False

```

You can control the timeouts and number of retries used when retrieving information from the Metadata Service (this is used for retrieving credentials for IAM roles on EC2 instances):

metadata_service_timeout Number of seconds until requests to the metadata service will timeout (float).

metadata_service_num_attempts Number of times to attempt to retrieve information from the metadata service before giving up (int).

These settings will default to:

```

[Boto]
metadata_service_timeout = 1.0
metadata_service_num_attempts = 1

```

This section is also used for specifying endpoints for non-AWS services such as Eucalyptus and Walrus.

eucalyptus_host Select a default endpoint host for eucalyptus

walrus_host Select a default host for Walrus

For example:

```

[Boto]
eucalyptus_host = somehost.example.com
walrus_host = somehost.example.com

```

Finally, the Boto section is used to set defaults versions for many AWS services

AutoScale settings:

options: :autoscale_version: Set the API version :autoscale_endpoint: Endpoint to use :autoscale_region_name: Default region to use

For example:

```

[Boto]
autoscale_version = 2011-01-01
autoscale_endpoint = autoscaling.us-west-2.amazonaws.com
autoscale_region_name = us-west-2

```

Cloudformation settings can also be defined:

cfn_version Cloud formation API version

cfn_region_name Default region name

cfn_region_endpoint Default endpoint

For example:

```

[Boto]
cfn_version = 2010-05-15
cfn_region_name = us-west-2
cfn_region_endpoint = cloudformation.us-west-2.amazonaws.com

```

Cloudsearch settings:

cs_region_name Default cloudsearch region

cs_region_endpoint Default cloudsearch endpoint

For example:

```
[Boto]
cs_region_name = us-west-2
cs_region_endpoint = cloudsearch.us-west-2.amazonaws.com
```

Cloudwatch settings:

cloudwatch_version Cloudwatch API version

cloudwatch_region_name Default region name

cloudwatch_region_endpoint Default endpoint

For example:

```
[Boto]
cloudwatch_version = 2010-08-01
cloudwatch_region_name = us-west-2
cloudwatch_region_endpoint = monitoring.us-west-2.amazonaws.com
```

EC2 settings:

ec2_version EC2 API version

ec2_region_name Default region name

ec2_region_endpoint Default endpoint

For example:

```
[Boto]
ec2_version = 2012-12-01
ec2_region_name = us-west-2
ec2_region_endpoint = ec2.us-west-2.amazonaws.com
```

ELB settings:

elb_version ELB API version

elb_region_name Default region name

elb_region_endpoint Default endpoint

For example:

```
[Boto]
elb_version = 2012-06-01
elb_region_name = us-west-2
elb_region_endpoint = elasticloadbalancing.us-west-2.amazonaws.com
```

EMR settings:

emr_version EMR API version

emr_region_name Default region name

emr_region_endpoint Default endpoint

For example:


```
[Boto]
emr_version = 2009-03-31
emr_region_name = us-west-2
emr_region_endpoint = elasticmapreduce.us-west-2.amazonaws.com
```

4.97.4 Precedence

Even if you have your boto config setup, you can also have credentials and options stored in environmental variables or you can explicitly pass them to method calls i.e.:

```
>>> boto.ec2.connect_to_region(
...     'us-west-2',
...     aws_access_key_id='foo',
...     aws_secret_access_key='bar')
```

In these cases where these options can be found in more than one place boto will first use the explicitly supplied arguments, if none found it will then look for them amidst environment variables and if that fails it will use the ones in boto config.

Notification

If you are using notifications for boto.pyami, you can specify the email details through the following variables.

smtp_from Used as the sender in notification emails.

smtp_to Destination to which emails should be sent

smtp_host Host to connect to when sending notification emails.

smtp_port Port to connect to when connecting to the :smtp_host:

Default values are:

```
[notification]
smtp_from = boto
smtp_to = None
smtp_host = localhost
smtp_port = 25
smtp_tls = True
smtp_user = john
smtp_pass = hunter2
```

SWF

The SWF section allows you to configure the default region to be used for the Amazon Simple Workflow service.

region Set the default region

Example:

```
[SWF]
region = us-west-2
```

Pyami

The Pyami section is used to configure the working directory for PyAMI.

working_dir Working directory used by PyAMI

Example:

```
[Pyami]
working_dir = /home/foo/
```

DB

The DB section is used to configure access to databases through the `boto.sdb.db.manager.get_manager()` function.

db_type Type of the database. Current allowed values are *SimpleDB* and *XML*.

db_user AWS access key id.

db_passwd AWS secret access key.

db_name Database that will be connected to.

db_table Table name :note: This doesn't appear to be used.

db_host Host to connect to

db_port Port to connect to

enable_ssl Use SSL

More examples:

```
[DB]
db_type = SimpleDB
db_user = <aws access key id>
db_passwd = <aws secret access key>
db_name = my_domain
db_table = table
db_host = sdb.amazonaws.com
enable_ssl = True
debug = True

[DB_TestBasic]
db_type = SimpleDB
db_user = <another aws access key id>
db_passwd = <another aws secret access key>
db_name = basic_domain
db_port = 1111
```

SDB

This section is used to configure SimpleDB

region Set the region to which SDB should connect

Example:

```
[SDB]
region = us-west-2
```

DynamoDB

This section is used to configure DynamoDB

region Choose the default region

validate_checksums Check checksums returned by DynamoDB

Example:

```
[DynamoDB]
region = us-west-2
validate_checksums = True
```

4.98 About the Documentation

boto's documentation uses the [Sphinx](#) documentation system, which in turn is based on [docutils](#). The basic idea is that lightly-formatted plain-text documentation is transformed into HTML, PDF, and any other output format.

To actually build the documentation locally, you'll currently need to install `Sphinx - easy_install Sphinx` should do the trick.

Then, building the html is easy; just make `html` from the `docs` directory.

To get started contributing, you'll want to read the [ReStructuredText Primer](#). After that, you'll want to read about the [Sphinx-specific markup](#) that's used to manage metadata, indexing, and cross-references.

The main thing to keep in mind as you write and edit docs is that the more semantic markup you can add the better. So:

```
Import ``boto`` to your script...
```

Isn't nearly as helpful as:

```
Add :mod:`boto` to your script...
```

This is because Sphinx will generate a proper link for the latter, which greatly helps readers. There's basically no limit to the amount of useful markup you can add.

4.98.1 The fabfile

There is a [Fabric](#) file that can be used to build and deploy the documentation to a webserver that you ssh access to.

To build and deploy:

```
cd docs/
fab deploy:remote_path='/var/www/folder/whatever' --hosts=user@host
```

This will get the latest code from subversion, add the revision number to the docs conf.py file, call `make html` to build the documentation, then it will tarball it up and scp up to the host you specified and untarball it in the folder you specified creating a symbolic link from the untarballled versioned folder to `{remote_path}/boto-docs`.

4.99 Contributing to Boto

4.99.1 Setting Up a Development Environment

While not strictly required, it is highly recommended to do development in a virtualenv. You can install virtualenv using pip:

```
$ pip install virtualenv
```

Once the package is installed, you'll have a `virtualenv` command you can use to create a virtual environment:

```
$ virtualenv venv
```

You can then activate the virtualenv:

```
$ . venv/bin/activate
```

Note: You may also want to check out [virtualenvwrapper](#), which is a set of extensions to virtualenv that makes it easy to manage multiple virtual environments.

A `requirements.txt` is included with `boto` which contains all the additional packages needed for `boto` development. You can install these packages by running:

```
$ pip install -r requirements.txt
```

4.99.2 Running the Tests

All of the tests for `boto` are under the `tests/` directory. The tests for `boto` have been split into two main categories, unit and integration tests:

- **unit** - These are tests that do not talk to any AWS services. Anyone should be able to run these tests without have any credentials configured. These are the types of tests that could be run in something like a public CI server. These tests tend to be fast.
- **integration** - These are tests that will talk to AWS services, and will typically require a `boto` config file with valid credentials. Due to the nature of these tests, they tend to take a while to run. Also keep in mind anyone who runs these tests will incur any usage fees associated with the various AWS services.

To run all the unit tests, `cd` to the `tests/` directory and run:

```
$ python test.py unit
```

You should see output like this:

```
$ python test.py unit
.....
-----
Ran 32 tests in 0.075s

OK
```

To run the integration tests, run:

```
$ python test.py integration
```

Note that running the integration tests may take a while.

Various integration tests have been tagged with service names to allow you to easily run tests by service type. For example, to run the ec2 integration tests you can run:

```
$ python test.py -t ec2
```

You can specify the `-t` argument multiple times. For example, to run the s3 and ec2 tests you can run:

```
$ python test.py -t ec2 -t s3
```

Warning: In the examples above no top level directory was specified. By default, nose will assume the current working directory, so the above command is equivalent to:

```
$ python test.py -t ec2 -t s3 .
```

Be sure that you are in the `tests/` directory when running the tests, or explicitly specify the top level directory. For example, if you in the root directory of the boto repo, you could run the ec2 and s3 tests by running:

```
$ python tests/test.py -t ec2 -t s3 tests/
```

You can use nose's collect plugin to see what tests are associated with each service tag:

```
$ python tests.py -t s3 -t ec2 --with-id --collect -v
```

Testing Details

The `tests/test.py` script is a lightweight wrapper around `nose`. In general, you should be able to run `nosetests` directly instead of `tests/test.py`. The `tests/unit` and `tests/integration` args in the commands above were referring to directories. The command line arguments are forwarded to nose when you use `tests/test.py`. For example, you can run:

```
$ python tests/test.py -x -vv tests/unit/cloudformation
```

And the `-x -vv tests/unit/cloudformation` are forwarded to nose. See the [nose docs](#) for the supported command line options, or run `nosetests --help`.

The only thing that `tests/test.py` does before invoking nose is to inject an argument that specifies that any testcase tagged with “notdefault” should not be run. A testcase may be tagged with “notdefault” if the test author does not want everyone to run the tests. In general, there shouldn't be many of these tests, but some reasons a test may be tagged “notdefault” include:

- An integration test that requires specific credentials.
- An interactive test (the S3 MFA tests require you to type in the S/N and code).

Tagging is done using nose's [tagging](#) plugin. To summarize, you can tag a specific testcase by setting an attribute on the object. Nose provides an `attr` decorator for convenience:

```
from nose.plugins.attrib import attr

@attr('notdefault')
```

(continues on next page)

(continued from previous page)

```
def test_s3_mfs():
    pass
```

You can then run these tests by specifying:

```
nosetests -a 'notdefault'
```

Or you can exclude any tests tagged with 'notdefault' by running:

```
nosetests -a '!notdefault'
```

Conceptually, `tests/test.py` is injecting the “-a !notdefault” arg into `nosetests`.

4.99.3 Testing Supported Python Versions

Boto supports python 2.6 and 2.7. An easy way to verify functionality across multiple python versions is to use `tox`. A `tox.ini` file is included with boto. You can run `tox` with no args and it will automatically test all supported python versions:

```
$ tox
GLOB sdist-make: boto/setup.py
py26 sdist-reinst: boto/.tox/dist/boto-2.4.1.zip
py26 runtests: commands[0]
.....
-----
Ran 32 tests in 0.089s

OK
py27 sdist-reinst: boto/.tox/dist/boto-2.4.1.zip
py27 runtests: commands[0]
.....
-----
Ran 32 tests in 0.087s

OK
____ summary ____
  py26: commands succeeded
  py27: commands succeeded
  congratulations :)
```

4.99.4 Writing Documentation

The boto docs use `sphinx` to generate documentation. All of the docs are located in the `docs/` directory. To generate the html documentation, cd into the docs directory and run `make html`:

```
$ cd docs
$ make html
```

The generated documentation will be in the `docs/build/html` directory. The source for the documentation is located in `docs/source` directory, and uses `restructured text` for the markup language.

4.99.5 Merging A Branch (Core Devs)

- All features/bugfixes should go through a review.
 - This includes new features added by core devs themselves. The usual branch/pull-request/merge flow that happens for community contributions should also apply to core.
- Ensure there is proper test coverage. If there's a change in behavior, there should be a test demonstrating the failure before the change & passing with the change.
 - This helps ensure we don't regress in the future as well.
- Merging of pull requests is typically done with `git merge --no-ff <remote/branch_name>`.
 - GitHub's big green button is probably OK for very small PRs (like doc fixes), but you can't run tests on GH, so most things should get pulled down locally.

4.100 Command Line Tools

4.100.1 Introduction

Boto ships with a number of command line utilities, which are installed when the package is installed. This guide outlines which ones are available & what they do.

Note: If you're not already depending on these utilities, you may wish to check out the AWS-CLI (<http://aws.amazon.com/cli/> - [User Guide](#) & [Reference Guide](#)). It provides much wider & complete access to the AWS services.

The included utilities available are:

asadmin Works with Autoscaling

bundle_image Creates a bundled AMI in S3 based on a EC2 instance

cfadmin Works with CloudFront & invalidations

cq Works with SQS queues

cwutil Works with CloudWatch

`dynamodb_dump` `dynamodb_load`

Handle dumping/loading data from DynamoDB tables

elbadmin Manages Elastic Load Balancer instances

fetch_file Downloads an S3 key to disk

glacier Lists vaults, jobs & uploads files to Glacier

instance_events Lists all events for EC2 reservations

kill_instance Kills a list of EC2 instances

launch_instance Launches an EC2 instance

list_instances Lists all of your EC2 instances

lss3 Lists what keys you have within a bucket in S3

mturk Provides a number of facilities for interacting with Mechanical Turk

pyami_sendmail Sends an email from the Pyami instance

route53 Interacts with the Route53 service

s3put Uploads a directory or a specific file(s) to S3

sdbadmin Allows for working with SimpleDB domains

taskadmin A tool for working with the tasks in SimpleDB

4.101 An Introduction to boto's Support interface

This tutorial focuses on the boto interface to Amazon Web Services Support, allowing you to programmatically interact with cases created with Support. This tutorial assumes that you have already downloaded and installed `boto`.

4.101.1 Creating a Connection

The first step in accessing Support is to create a connection to the service. There are two ways to do this in boto. The first is:

```
>>> from boto.support.connection import SupportConnection
>>> conn = SupportConnection('<aws access key>', '<aws secret key>')
```

At this point the variable `conn` will point to a `SupportConnection` object. In this example, the AWS access key and AWS secret key are passed in to the method explicitly. Alternatively, you can set the environment variables:

AWS_ACCESS_KEY_ID Your AWS Access Key ID

AWS_SECRET_ACCESS_KEY Your AWS Secret Access Key

and then call the constructor without any arguments, like this:

```
>>> conn = SupportConnection()
```

There is also a shortcut function in boto that makes it easy to create Support connections:

```
>>> import boto.support
>>> conn = boto.support.connect_to_region('us-west-2')
```

In either case, `conn` points to a `SupportConnection` object which we will use throughout the remainder of this tutorial.

4.101.2 Describing Existing Cases

If you have existing cases or want to fetch cases in the future, you'll use the `SupportConnection.describe_cases` method. For example:

```
>>> cases = conn.describe_cases()
>>> len(cases['cases'])
1
>>> cases['cases'][0]['title']
'A test case.'
>>> cases['cases'][0]['caseId']
'case-...'
```

You can also fetch a set of cases (or single case) by providing a `case_id_list` parameter:


```
>>> cases = conn.describe_cases(case_id_list=['case-1'])
>>> len(cases['cases'])
1
>>> cases['cases'][0]['title']
'A test case.'
>>> cases['cases'][0]['caseId']
'case-...'
```

4.101.3 Describing Service Codes

In order to create a new case, you'll need to fetch the service (& category) codes available to you. Fetching them is a simple call to:

```
>>> services = conn.describe_services()
>>> services['services'][0]['code']
'amazon-cloudsearch'
```

If you only care about certain services, you can pass a list of service codes:

```
>>> service_details = conn.describe_services(service_code_list=[
...     'amazon-cloudsearch',
...     'amazon-dynamodb',
... ])
```

4.101.4 Describing Severity Levels

In order to create a new case, you'll also need to fetch the severity levels available to you. Fetching them looks like:

```
>>> severities = conn.describe_severity_levels()
>>> severities['severityLevels'][0]['code']
'low'
```

4.101.5 Creating a Case

Upon creating a connection to Support, you can now work with existing Support cases, create new cases or resolve them. We'll start with creating a new case:

```
>>> new_case = conn.create_case(
...     subject='This is a test case.',
...     service_code='',
...     category_code='',
...     communication_body="",
...     severity_code='low'
... )
>>> new_case['caseId']
'case-...'
```

For the `service_code/category_code` parameters, you'll need to do a `SupportConnection.describe_services` call, then select the appropriate service code (& appropriate category code within that service) from the response.

For the `severity_code` parameter, you'll need to do a `SupportConnection.describe_severity_levels` call, then select the appropriate severity code from the response.

4.101.6 Adding to a Case

Since the purpose of a support case involves back-and-forth communication, you can add additional communication to the case as well. Providing a response might look like:

```
>>> result = conn.add_communication_to_case(
...     communication_body="This is a followup. It's working now."
...     case_id='case-...'
... )
```

4.101.7 Fetching all Communications for a Case

Getting all communications for a given case looks like:

```
>>> communications = conn.describe_communications('case-...')
```

4.101.8 Resolving a Case

Once a case is finished, you should mark it as resolved to close it out. Resolving a case looks like:

```
>>> closed = conn.resolve_case(case_id='case-...')
>>> closed['result']
True
```

4.102 An Introduction to boto's DynamoDB v2 interface

This tutorial focuses on the boto interface to AWS' [DynamoDB v2](#). This tutorial assumes that you have boto already downloaded and installed.

Warning: This tutorial covers the **SECOND** major release of DynamoDB (including local secondary index support). The documentation for the original version of DynamoDB (& boto's support for it) is at [DynamoDB v1](#).

The v2 DynamoDB API has both a high-level & low-level component. The low-level API (contained primarily within `boto.dynamodb2.layer1`) provides an interface that rough matches exactly what is provided by the API. It supports all options available to the service.

The high-level API attempts to make interacting with the service more natural from Python. It supports most of the featureset.

4.102.1 The High-Level API

Most of the interaction centers around a single object, the `Table`. Tables act as a way to effectively namespace your records. If you're familiar with database tables from an RDBMS, tables will feel somewhat familiar.

Creating a New Table

To create a new table, you need to call `Table.create` & specify (at a minimum) both the table's name as well as the key schema for the table:

```
>>> from boto.dynamodb2.fields import HashKey
>>> from boto.dynamodb2.table import Table
>>> users = Table.create('users', schema=[HashKey('username')]);
```

Since both the key schema and local secondary indexes can not be modified after the table is created, you'll need to plan ahead of time how you think the table will be used. Both the keys & indexes are also used for querying, so you'll want to represent the data you'll need when querying there as well.

For the schema, you can either have a single `HashKey` or a combined `HashKey+RangeKey`. The `HashKey` by itself should be thought of as a unique identifier (for instance, like a username or UUID). It is typically looked up as an exact value. A `HashKey+RangeKey` combination is slightly different, in that the `HashKey` acts like a namespace/prefix & the `RangeKey` acts as a value that can be referred to by a sorted range of values.

For the local secondary indexes, you can choose from an `AllIndex`, a `KeysOnlyIndex` or a `IncludeIndex` field. Each builds an index of values that can be queried on. The `AllIndex` duplicates all values onto the index (to prevent additional reads to fetch the data). The `KeysOnlyIndex` duplicates only the keys from the schema onto the index. The `IncludeIndex` lets you specify a list of fieldnames to duplicate over.

A full example:

```
>>> import boto.dynamodb2
>>> from boto.dynamodb2.fields import HashKey, RangeKey, KeysOnlyIndex, GlobalAllIndex
>>> from boto.dynamodb2.table import Table
>>> from boto.dynamodb2.types import NUMBER

# Uses your ``aws_access_key_id`` & ``aws_secret_access_key`` from either a
# config file or environment variable & the default region.
>>> users = Table.create('users', schema=[
...     HashKey('username'), # defaults to STRING data_type
...     RangeKey('last_name'),
... ], throughput={
...     'read': 5,
...     'write': 15,
... }, global_indexes=[
...     GlobalAllIndex('EverythingIndex', parts=[
...         HashKey('account_type'),
...     ],
...     throughput={
...         'read': 1,
...         'write': 1,
...     })
... ],
... # If you need to specify custom parameters, such as credentials or region,
... # use the following:
... # connection=boto.dynamodb2.connect_to_region('us-east-1')
... )
```

Using an Existing Table

Once a table has been created, using it is relatively simple. You can either specify just the `table_name` (allowing the object to lazily do an additional call to get details about itself if needed) or provide the `schema/indexes` again (same as what was used with `Table.create`) to avoid extra overhead.

Lazy example:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')
```

Efficient example:

```
>>> from boto.dynamodb2.fields import HashKey, RangeKey, GlobalAllIndex
>>> from boto.dynamodb2.table import Table
>>> from boto.dynamodb2.types import NUMBER
>>> users = Table('users', schema=[
...     HashKey('username'),
...     RangeKey('last_name'),
... ], global_indexes=[
...     GlobalAllIndex('EverythingIndex', parts=[
...         HashKey('account_type'),
...     ])
... ])
```

Creating a New Item

Once you have a `Table` instance, you can add new items to the table. There are two ways to do this.

The first is to use the `Table.put_item` method. Simply hand it a dictionary of data & it will create the item on the server side. This dictionary should be relatively flat (as you can nest in other dictionaries) & **must** contain the keys used in the schema.

Example:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

# Create the new user.
>>> users.put_item(data={
...     'username': 'johndoe',
...     'first_name': 'John',
...     'last_name': 'Doe',
...     'account_type': 'standard_user',
... })
True
```

The alternative is to manually construct an `Item` instance & tell it to save itself. This is useful if the object will be around for awhile & you don't want to re-fetch it.

Example:

```
>>> from boto.dynamodb2.items import Item
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

# WARNING - This doesn't save it yet!
>>> janedoe = Item(users, data={
...     'username': 'janedoe',
...     'first_name': 'Jane',
...     'last_name': 'Doe',
...     'account_type': 'standard_user',
... })

# The data now gets persisted to the server.
>>> janedoe.save()
True
```

Getting an Item & Accessing Data

With data now in DynamoDB, if you know the key of the item, you can fetch it back out. Specify the key value(s) as kwargs to `Table.get_item`.

Example:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

>>> johndoe = users.get_item(username='johndoe', last_name='Doe')
```

Once you have an `Item` instance, it presents a dictionary-like interface to the data.:

```
# Read a field out.
>>> johndoe['first_name']
'John'

# Change a field (DOESN'T SAVE YET!).
>>> johndoe['first_name'] = 'Johann'

# Delete data from it (DOESN'T SAVE YET!).
>>> del johndoe['account_type']
```

Updating an Item

Just creating new items or changing only the in-memory version of the `Item` isn't particularly effective. To persist the changes to DynamoDB, you have three choices.

The first is sending all the data with the expectation nothing has changed since you read the data. DynamoDB will verify the data is in the original state and, if so, will send all of the item's data. If that expectation fails, the call will fail:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

>>> johndoe = users.get_item(username='johndoe', last_name='Doe')
>>> johndoe['first_name'] = 'Johann'
>>> johndoe['whatever'] = "man, that's just like your opinion"
>>> del johndoe['account_type']

# Affects all fields, even the ones not changed locally.
>>> johndoe.save()
True
```

The second is a full overwrite. If you can be confident your version of the data is the most correct, you can force an overwrite of the data.:

```
>>> johndoe = users.get_item(username='johndoe', last_name='Doe')
>>> johndoe['first_name'] = 'Johann'
>>> johndoe['whatever'] = "Man, that's just like your opinion"

# Specify ``overwrite=True`` to fully replace the data.
>>> johndoe.save(overwrite=True)
True
```

The last is a partial update. If you've only modified certain fields, you can send a partial update that only writes those fields, allowing other (potentially changed) fields to go untouched.:

```
>>> johndoe = users.get_item(username='johndoe', last_name='Doe')
>>> johndoe['first_name'] = 'Johann'
>>> johndoe['whatever'] = "man, that's just like your opinion"
>>> del johndoe['account_type']

# Partial update, only sending/affecting the
# ``first_name/whatever/account_type`` fields.
>>> johndoe.partial_save()
True
```

Deleting an Item

You can also delete items from the table. You have two choices, depending on what data you have present.

If you already have an `Item` instance, the easiest approach is just to call `Item.delete`:

```
>>> johndoe.delete()
True
```

If you don't have an `Item` instance & you don't want to incur the `Table.get_item` call to get it, you can call `Table.delete_item` method.:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

>>> users.delete_item(username='johndoe', last_name='Doe')
True
```

Batch Writing

If you're loading a lot of data at a time, making use of batch writing can both speed up the process & reduce the number of write requests made to the service.

Batch writing involves wrapping the calls you want batched in a context manager. The context manager imitates the `Table.put_item` & `Table.delete_item` APIs. Getting & using the context manager looks like:

```
>>> import time
>>> from boto.dynamodb2.table import Table
>>> users = Table('users')

>>> with users.batch_write() as batch:
...     batch.put_item(data={
...         'username': 'anotherdoe',
...         'first_name': 'Another',
...         'last_name': 'Doe',
...         'date_joined': int(time.time()),
...     })
...     batch.put_item(data={
...         'username': 'joebloggs',
...         'first_name': 'Joe',
...         'last_name': 'Bloggs',
...         'date_joined': int(time.time()),
```

(continues on next page)

(continued from previous page)

```
...     })
...     batch.delete_item(username='janedoe', last_name='Doe')
```

However, there are some limitations on what you can do within the context manager.

- It can't read data at all or do batch any other operations.
- You can't put & delete the same data within a batch request.

Note: Additionally, the context manager can only batch 25 items at a time for a request (this is a DynamoDB limitation). It is handled for you so you can keep writing additional items, but you should be aware that 100 `put_item` calls is 4 batch requests, not 1.

Querying

Warning: The `Table` object has both a `query` & a `query_2` method. If you are writing new code, **DO NOT** use `Table.query`. It presents results in an incorrect order than expected & is strictly present for backward-compatibility.

Manually fetching out each item by itself isn't tenable for large datasets. To cope with fetching many records, you can either perform a standard query, query via a local secondary index or scan the entire table.

A standard query typically gets run against a hash+range key combination. Filter parameters are passed as kwargs & use a `__` to separate the fieldname from the operator being used to filter the value.

In terms of querying, our original schema is less than optimal. For the following examples, we'll be using the following table setup:

```
>>> from boto.dynamodb2.fields import HashKey, RangeKey, GlobalAllIndex
>>> from boto.dynamodb2.table import Table
>>> from boto.dynamodb2.types import NUMBER
>>> import time
>>> users = Table.create('users2', schema=[
...     HashKey('account_type'),
...     RangeKey('last_name'),
... ], throughput={
...     'read': 5,
...     'write': 15,
... }, global_indexes=[
...     GlobalAllIndex('DateJoinedIndex', parts=[
...         HashKey('account_type'),
...         RangeKey('date_joined', data_type=NUMBER),
...     ],
...     throughput={
...         'read': 1,
...         'write': 1,
...     })),
... 1)
```

And the following data:

```

>>> with users.batch_write() as batch:
...     batch.put_item(data={
...         'account_type': 'standard_user',
...         'first_name': 'John',
...         'last_name': 'Doe',
...         'is_owner': True,
...         'email': True,
...         'date_joined': int(time.time()) - (60*60*2),
...     })
...     batch.put_item(data={
...         'account_type': 'standard_user',
...         'first_name': 'Jane',
...         'last_name': 'Doering',
...         'date_joined': int(time.time()) - 2,
...     })
...     batch.put_item(data={
...         'account_type': 'standard_user',
...         'first_name': 'Bob',
...         'last_name': 'Doerr',
...         'date_joined': int(time.time()) - (60*60*3),
...     })
...     batch.put_item(data={
...         'account_type': 'super_user',
...         'first_name': 'Alice',
...         'last_name': 'Liddel',
...         'is_owner': True,
...         'email': True,
...         'date_joined': int(time.time()) - 1,
...     })

```

When executing the query, you get an iterable back that contains your results. These results may be spread over multiple requests as DynamoDB paginates them. This is done transparently, but you should be aware it may take more than one request.

To run a query for last names starting with the letter “D”:

```

>>> names_with_d = users.query_2(
...     account_type__eq='standard_user',
...     last_name__startswith='D'
... )

>>> for user in names_with_d:
...     print user['first_name']
'John'
'Jane'
'Bob'

```

You can also reverse results (`reverse=True`) as well as limiting them (`limit=2`):

```

>>> rev_with_d = users.query_2(
...     account_type__eq='standard_user',
...     last_name__startswith='D',
...     reverse=True,
...     limit=2
... )

>>> for user in rev_with_d:

```

(continues on next page)

(continued from previous page)

```
...     print user['first_name']
'Bob'
'Jane'
```

You can also run queries against the local secondary indexes. Simply provide the index name (index='DateJoinedIndex') & filter parameters against its fields:

```
# Users within the last hour.
>>> recent = users.query_2(
...     account_type__eq='standard_user',
...     date_joined__gte=time.time() - (60 * 60),
...     index='DateJoinedIndex'
... )

>>> for user in recent:
...     print user['first_name']
'Jane'
```

By default, DynamoDB can return a large amount of data per-request (up to 1Mb of data). To prevent these requests from drowning other smaller gets, you can specify a smaller page size via the `max_page_size` argument to `Table.query_2` & `Table.scan`. Doing so looks like:

```
# Small pages yield faster responses & less potential of drowning other
# requests.
>>> all_users = users.query_2(
...     account_type__eq='standard_user',
...     date_joined__gte=0,
...     index='DateJoinedIndex',
...     max_page_size=10
... )

# Usage is the same, but now many smaller requests are done.
>>> for user in all_users:
...     print user['first_name']
'Bob'
'John'
'Jane'
```

Finally, if you need to query on data that's not in either a key or in an index, you can run a `Table.scan` across the whole table, which accepts a similar but expanded set of filters. If you're familiar with the Map/Reduce concept, this is akin to what DynamoDB does.

Warning: Scans are eventually consistent & run over the entire table, so relatively speaking, they're more expensive than plain queries or queries against an LSI.

An example scan of all records in the table looks like:

```
>>> all_users = users.scan()
```

Filtering a scan looks like:

```
>>> owners_with_emails = users.scan(
...     is_owner__eq=True,
...     email__null=False,
```

(continues on next page)

(continued from previous page)

```
... )  
  
>>> for user in owners_with_emails:  
...     print user['first_name']  
'John'  
'Alice'
```

The ResultSet

Both `Table.query_2` & `Table.scan` return an object called `ResultSet`. It's a lazily-evaluated object that uses the `Iterator protocol`. It delays your queries until you request the next item in the result set.

Typical use is simply a standard `for` to iterate over the results:

```
>>> result_set = users.scan()  
>>> for user in result_set:  
...     print user['first_name']  
'John'  
'Jane'  
'Bob'  
'Alice'
```

However, this throws away results as it fetches more data. As a result, you can't index it like a list:

```
>>> len(result_set)  
TypeError: object of type 'ResultSet' has no len()
```

Because it does this, if you need to loop over your results more than once (or do things like negative indexing, length checks, etc.), you should wrap it in a call to `list()`. Ex.:

```
>>> result_set = users.scan()  
>>> all_users = list(result_set)  
# Slice it for every other user.  
>>> for user in all_users[::2]:  
...     print user['first_name']  
'John'  
'Bob'
```

Warning: Wrapping calls like the above in `list(...)` **WILL** cause it to evaluate the **ENTIRE** potentially large data set.

Appropriate use of the `limit=...` kwarg to `Table.query_2` & `Table.scan` calls are **VERY** important should you choose to do this.

Alternatively, you can build your own list, using `for` on the `ResultSet` to lazily build the list (& potentially stop early).

Parallel Scan

DynamoDB also includes a feature called “Parallel Scan”, which allows you to make use of **extra** read capacity to divide up your result set & scan an entire table faster.

This does require extra code on the user's part & you should ensure that you need the speed boost, have enough data to justify it and have the extra capacity to read it without impacting other queries/scans.

To run it, you should pick the `total_segments` to use, which is an integer representing the number of temporary partitions you'd divide your table into. You then need to spin up a thread/process for each one, giving each thread/process a segment, which is a zero-based integer of the segment you'd like to scan.

An example of using parallel scan to send out email to all users might look something like:

```
#!/usr/bin/env python
import threading

import boto.ses
import boto.dynamodb2
from boto.dynamodb2.table import Table

AWS_ACCESS_KEY_ID = '<YOUR_AWS_KEY_ID>'
AWS_SECRET_ACCESS_KEY = '<YOUR_AWS_SECRET_KEY>'
APPROVED_EMAIL = 'some@address.com'

def send_email(email):
    # Using Amazon's Simple Email Service, send an email to a given
    # email address. You must already have an email you've verified with
    # AWS before this will work.
    conn = boto.ses.connect_to_region(
        'us-east-1',
        aws_access_key_id=AWS_ACCESS_KEY_ID,
        aws_secret_access_key=AWS_SECRET_ACCESS_KEY
    )
    conn.send_email(
        APPROVED_EMAIL,
        "[OurSite] New feature alert!",
        "We've got some exciting news! We added a new feature to...",
        [email]
    )

def process_segment(segment=0, total_segments=10):
    # This method/function is executed in each thread, each getting its
    # own segment to process through.
    conn = boto.dynamodb2.connect_to_region(
        'us-east-1',
        aws_access_key_id=AWS_ACCESS_KEY_ID,
        aws_secret_access_key=AWS_SECRET_ACCESS_KEY
    )
    table = Table('users', connection=conn)

    # We pass in the segment & total_segments to scan here.
    for user in table.scan(segment=segment, total_segments=total_segments):
        send_email(user['email'])

def send_all_emails():
    pool = []
    # We're choosing to divide the table in 3, then...
    pool_size = 3
```

(continues on next page)

(continued from previous page)

```
# ...spinning up a thread for each segment.
for i in range(pool_size):
    worker = threading.Thread(
        target=process_segment,
        kwargs={
            'segment': i,
            'total_segments': pool_size,
        }
    )
    pool.append(worker)
    # We start them to let them start scanning & consuming their
    # assigned segment.
    worker.start()

# Finally, we wait for each to finish.
for thread in pool:
    thread.join()

if __name__ == '__main__':
    send_all_emails()
```

Batch Reading

Similar to batch writing, batch reading can also help reduce the number of API requests necessary to access a large number of items. The `Table.batch_get` method takes a list (or any sliceable collection) of keys & fetches all of them, presented as an iterator interface.

This is done lazily, so if you never iterate over the results, no requests are executed. Additionally, if you only iterate over part of the set, the minimum number of calls are made to fetch those results (typically max 100 per response).

Example:

```
>>> from boto.dynamodb2.table import Table
>>> users = Table('users2')

# No request yet.
>>> many_users = users.batch_get(keys=[
...     {'account_type': 'standard_user', 'last_name': 'Doe'},
...     {'account_type': 'standard_user', 'last_name': 'Doering'},
...     {'account_type': 'super_user', 'last_name': 'Liddel'},
... ])

# Now the request is performed, requesting all five in one request.
>>> for user in many_users:
...     print user['first_name']
'Alice'
'John'
'Jane'
```

Deleting a Table

Deleting a table is a simple exercise. When you no longer need a table, simply run:

```
>>> users.delete()
```

DynamoDB Local

Amazon DynamoDB Local is a utility which can be used to mock DynamoDB during development. Connecting to a running DynamoDB Local server is easy:

```
#!/usr/bin/env python
from boto.dynamodb2.layer1 import DynamoDBConnection

# Connect to DynamoDB Local
conn = DynamoDBConnection(
    host='localhost',
    port=8000,
    aws_access_key_id='anything',
    aws_secret_access_key='anything',
    is_secure=False)

# List all local tables
tables = conn.list_tables()
```

Next Steps

You can find additional information about other calls & parameter options in the [API docs](#).

4.103 Migrating from DynamoDB v1 to DynamoDB v2

For the v2 release of AWS' `DynamoDB`, the high-level API for interacting via `boto` was rewritten. Since there were several new features added in v2, people using the v1 API may wish to transition their code to the new API. This guide covers the high-level APIs.

4.103.1 Creating New Tables

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> message_table_schema = conn.create_schema(
...     hash_key_name='forum_name',
...     hash_key_proto_value=str,
...     range_key_name='subject',
...     range_key_proto_value=str
... )
>>> table = conn.create_table(
...     name='messages',
...     schema=message_table_schema,
...     read_units=10,
...     write_units=10
... )
```

DynamoDB v2:

```
>>> from boto.dynamodb2.fields import HashKey
>>> from boto.dynamodb2.fields import RangeKey
>>> from boto.dynamodb2.table import Table

>>> table = Table.create('messages', schema=[
...     HashKey('forum_name'),
...     RangeKey('subject'),
... ], throughput={
...     'read': 10,
...     'write': 10,
... })
```

4.103.2 Using an Existing Table

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
# With API calls.
>>> table = conn.get_table('messages')

# Without API calls.
>>> message_table_schema = conn.create_schema(
...     hash_key_name='forum_name',
...     hash_key_proto_value=str,
...     range_key_name='subject',
...     range_key_proto_value=str
... )
>>> table = conn.table_from_schema(
...     name='messages',
...     schema=message_table_schema)
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
# With API calls.
>>> table = Table('messages')

# Without API calls.
>>> from boto.dynamodb2.fields import HashKey
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages', schema=[
...     HashKey('forum_name'),
...     HashKey('subject'),
... ])
```

4.103.3 Updating Throughput

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
```

(continues on next page)

(continued from previous page)

```
>>> table = conn.get_table('messages')
>>> conn.update_throughput(table, read_units=5, write_units=15)
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')
>>> table.update(throughput={
...     'read': 5,
...     'write': 15,
... })
```

4.103.4 Deleting a Table

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')
>>> conn.delete_table(table)
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')
>>> table.delete()
```

4.103.5 Creating an Item

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')
>>> item_data = {
...     'Body': 'http://url_to_lolcat.gif',
...     'SentBy': 'User A',
...     'ReceivedTime': '12/9/2011 11:36:03 PM',
... }
>>> item = table.new_item(
...     # Our hash key is 'forum'
...     hash_key='LOLCat Forum',
...     # Our range key is 'subject'
...     range_key='Check this out!',
...     # This has the
...     attrs=item_data
... )
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')
>>> item = table.put_item(data={
```

(continues on next page)

(continued from previous page)

```
...     'forum_name': 'LOLCat Forum',
...     'subject': 'Check this out!',
...     'Body': 'http://url_to_lolcat.gif',
...     'SentBy': 'User A',
...     'ReceivedTime': '12/9/2011 11:36:03 PM',
... })
```

4.103.6 Getting an Existing Item

DynamoDB v1:

```
>>> table = conn.get_table('messages')
>>> item = table.get_item(
...     hash_key='LOLCat Forum',
...     range_key='Check this out!'
... )
```

DynamoDB v2:

```
>>> table = Table('messages')
>>> item = table.get_item(
...     forum_name='LOLCat Forum',
...     subject='Check this out!'
... )
```

4.103.7 Updating an Item

DynamoDB v1:

```
>>> item['a_new_key'] = 'testing'
>>> del item['a_new_key']
>>> item.put()
```

DynamoDB v2:

```
>>> item['a_new_key'] = 'testing'
>>> del item['a_new_key']

# Conditional save, only if data hasn't changed.
>>> item.save()

# Forced full overwrite.
>>> item.save(overwrite=True)

# Partial update (only changed fields).
>>> item.partial_save()
```

4.103.8 Deleting an Item

DynamoDB v1:


```
>>> item.delete()
```

DynamoDB v2:

```
>>> item.delete()
```

4.103.9 Querying

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')
>>> from boto.dynamodb.condition import BEGINS_WITH
>>> items = table.query('Amazon DynamoDB',
...                     range_key_condition=BEGINS_WITH('DynamoDB'),
...                     request_limit=1, max_results=1)
>>> for item in items:
>>>     print item['Body']
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')
>>> items = table.query_2(
...     forum_name__eq='Amazon DynamoDB',
...     subject__beginswith='DynamoDB',
...     limit=1
... )
>>> for item in items:
>>>     print item['Body']
```

4.103.10 Scans

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')

# All items.
>>> items = table.scan()

# With a filter.
>>> items = table.scan(scan_filter={'Replies': GT(0)})
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')

# All items.
>>> items = table.scan()
```

(continues on next page)

(continued from previous page)

```
# With a filter.
>>> items = table.scan(replies__gt=0)
```

4.103.11 Batch Gets

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')
>>> from boto.dynamodb.batch import BatchList
>>> the_batch = BatchList(conn)
>>> the_batch.add_batch(table, keys=[
...     ('LOLCat Forum', 'Check this out!'),
...     ('LOLCat Forum', 'I can haz docs?'),
...     ('LOLCat Forum', 'Maru'),
... ])
>>> results = conn.batch_get_item(the_batch)

# (Largely) Raw dictionaries back from DynamoDB.
>>> for item_dict in response['Responses'][table.name]['Items']:
...     print item_dict['Body']
```

DynamoDB v2:

```
>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')
>>> results = table.batch_get(keys=[
...     {'forum_name': 'LOLCat Forum', 'subject': 'Check this out!'},
...     {'forum_name': 'LOLCat Forum', 'subject': 'I can haz docs?'},
...     {'forum_name': 'LOLCat Forum', 'subject': 'Maru'},
... ])

# Lazy requests across pages, if paginated.
>>> for res in results:
...     # You get back actual ``Item`` instances.
...     print item['Body']
```

4.103.12 Batch Writes

DynamoDB v1:

```
>>> import boto.dynamodb
>>> conn = boto.dynamodb.connect_to_region()
>>> table = conn.get_table('messages')
>>> from boto.dynamodb.batch import BatchWriteList
>>> from boto.dynamodb.item import Item

# You must manually manage this so that your total ``puts/deletes`` don't
# exceed 25.
>>> the_batch = BatchList(conn)
>>> the_batch.add_batch(table, puts=[
...     Item(table, 'Corgi Fanciers', 'Sploots!', {
```

(continues on next page)

(continued from previous page)

```

...     'Body': 'Post your favorite corgi-on-the-floor shots!',
...     'SentBy': 'User B',
...     'ReceivedTime': '2013/05/02 10:56:45 AM',
...     }),
...     Item(table, 'Corgi Fanciers', 'Maximum FRAPS', {
...         'Body': 'http://internetvideosite/watch?v=1247869',
...         'SentBy': 'User C',
...         'ReceivedTime': '2013/05/01 09:15:25 PM',
...     }),
... ], deletes=[
...     ('LOLCat Forum', 'Off-topic post'),
...     ('LOLCat Forum', 'They be stealin mah bukket!'),
... ])
>>> conn.batch_write_item(the_writes)

```

DynamoDB v2:

```

>>> from boto.dynamodb2.table import Table
>>> table = Table('messages')

# Uses a context manager, which also automatically handles batch sizes.
>>> with table.batch_write() as batch:
...     batch.delete_item(
...         forum_name='LOLCat Forum',
...         subject='Off-topic post'
...     )
...     batch.put_item(data={
...         'forum_name': 'Corgi Fanciers',
...         'subject': 'Sploots!',
...         'Body': 'Post your favorite corgi-on-the-floor shots!',
...         'SentBy': 'User B',
...         'ReceivedTime': '2013/05/02 10:56:45 AM',
...     })
...     batch.put_item(data={
...         'forum_name': 'Corgi Fanciers',
...         'subject': 'Sploots!',
...         'Body': 'Post your favorite corgi-on-the-floor shots!',
...         'SentBy': 'User B',
...         'ReceivedTime': '2013/05/02 10:56:45 AM',
...     })
...     batch.delete_item(
...         forum_name='LOLCat Forum',
...         subject='They be stealin mah bukket!'
...     )

```

4.104 Migrating from RDS v1 to RDS v2

The original `boto.rds` module has historically lagged quite far behind the service (at time of writing, almost 50% of the API calls are missing/out-of-date). To address this, the Boto core team has switched to a generated client for RDS (`boto.rds2.layer1.RDSConnection`).

However, this generated variant is not backward-compatible with the older `boto.rds.RDSConnection`. This document is to help you update your code (as desired) to take advantage of the latest API calls.

For the duration of the document, **RDS2Connection** refers to `boto.rds2.layer1.RDSConnection`, where

RDSConnection refers to `boto.rds.RDSConnection`.

4.104.1 Prominent Differences

- The new **RDS2Connection** maps very closely to the [official API operations](#), where the old **RDSConnection** had non-standard & inconsistent method names.
- **RDS2Connection** almost always returns a Python dictionary that maps closely to the API output. **RDSConnection** returned Python objects.
- **RDS2Connection** is much more verbose in terms of output. Tools like [jmespath](#) or [jsonq](#) can make handling these sometimes complex dictionaries more manageable.

4.104.2 Method Renames

Format is `old_method_name -> new_method_name`:

- `authorize_dbsecurity_group -> authorize_db_security_group_ingress`
- `create_dbinstance -> create_db_instance`
- `create_dbinstance_read_replica -> create_db_instance_read_replica`
- `create_parameter_group -> create_db_parameter_group`
- `get_all_dbsnapshots -> describe_db_snapshots`
- `get_all_events -> describe_events`
- `modify_dbinstance -> modify_db_instance`
- `reboot_dbinstance -> reboot_db_instance`
- `restore_dbinstance_from_dbsnapshot -> restore_db_instance_from_db_snapshot`
- `restore_dbinstance_from_point_in_time -> restore_db_instance_to_point_in_time`
- `revoke_dbsecurity_group -> revoke_db_security_group_ingress`

4.104.3 Parameter Changes

Many parameter names have changed between **RDSConnection** & **RDS2Connection**. For instance, the old name for the instance identifier was `id`, where the new name is `db_instance_identifier`. These changes are to ensure things map more closely to the API.

In addition, in some cases, ordering & required-ness of parameters has changed as well. For instance, in `create_db_instance`, the `engine` parameter is now required (previously defaulted to `MySQL5.1`) & its position in the call has change to be before `master_username`.

As such, when updating your API calls, you should check the API Reference documentation to ensure you're passing the correct parameters.

4.104.4 Return Values

RDSConnection frequently returned higher-level Python objects. In contrast, **RDS2Connection** returns Python dictionaries of the data. This will require a bit more work to extract the necessary values. For example:

```

# Old
>>> instances = rds1_conn.get_all_dbinstances()
>>> inst = instances[0]
>>> inst.name
'test-db'

# New
>>> instances = rds2_conn.describe_db_instances()
>>> inst = instances['DescribeDBInstancesResponse']\
...           ['DescribeDBInstancesResult']['DBInstances'][0]
>>> inst['DBName']
'test-db'

```

4.105 Applications Built On Boto

Many people have taken Boto and layered on additional functionality, then shared them with the community. This is a (partial) list of applications that use Boto.

If you have an application or utility you've open-sourced that uses Boto & you'd like it listed here, please submit a [pull request](#) adding it!

botornado <https://pypi.python.org/pypi/botornado> An asynchronous AWS client on Tornado. This is a dirty work to move boto onto Tornado ioloop. Currently works with SQS and S3.

boto_rsync https://pypi.python.org/pypi/boto_rsync boto-rsync is a rough adaptation of boto's s3put script which has been reengineered to more closely mimic rsync. Its goal is to provide a familiar rsync-like wrapper for boto's S3 and Google Storage interfaces.

boto_utils https://pypi.python.org/pypi/boto_utils Command-line tools for interacting with Amazon Web Services, based on Boto. Includes utils for S3, SES & Cloudwatch.

django-storages <https://pypi.python.org/pypi/django-storages> A collection of storage backends for Django. Features the `S3BotoStorage` backend for storing media on S3.

mr.awesome <https://pypi.python.org/pypi/mr.awesome> mr.awesome is a commandline-tool (aws) to manage and control Amazon Webservice's EC2 instances. Once configured with your AWS key, you can create, delete, monitor and ssh into instances, as well as perform scripted tasks on them (via fabfiles). Examples are adding additional, pre-configured webservers to a cluster (including updating the load balancer), performing automated software deployments and creating backups - each with just one call from the commandline.

iamer <https://pypi.python.org/pypi/iamer> IAMer dump and load your AWS IAM configuration into text files. Once dumped, you can version the resulting json and ini files to keep track of changes, and even ask your team mates to do Pull Requests when they want access to something.

4.106 Auto Scaling Reference

4.106.1 boto.ec2.autoscale

This module provides an interface to the Elastic Compute Cloud (EC2) Auto Scaling service.

```
class boto.ec2.autoscale.AutoScaleConnection (aws_access_key_id=None,
                                             aws_secret_access_key=None,
                                             is_secure=True, port=None, proxy=None,
                                             proxy_port=None, proxy_user=None,
                                             proxy_pass=None, debug=0,
                                             https_connection_factory=None,
                                             region=None, path='/', security_token=None,
                                             validate_certs=True, profile_name=None,
                                             use_block_device_types=False)
```

Init method to create a new connection to the AutoScaling service.

B{Note:} The host argument is overridden by the host specified in the boto configuration file.

APIVersion = '2011-01-01'

DefaultRegionEndpoint = 'autoscaling.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

attach_instances (*name, instance_ids*)
 Attach instances to an autoscaling group.

build_list_params (*params, items, label*)
 Items is a list of dictionaries or strings:

```
[
  {
    'Protocol' : 'HTTP',
    'LoadBalancerPort' : '80',
    'InstancePort' : '80'
  },
  ..
] etc.
```

or:

```
['us-east-1b', ...]
```

create_auto_scaling_group (*as_group*)
 Create auto scaling group.

create_launch_configuration (*launch_config*)
 Creates a new Launch Configuration.

Parameters launch_config (*boto.ec2.autoscale.launchconfig.LaunchConfiguration*) – LaunchConfiguration object.

create_or_update_tags (*tags*)
 Creates new tags or updates existing tags for an Auto Scaling group.

Parameters tags (List of *boto.ec2.autoscale.tag.Tag*) – The new or updated tags.

create_scaling_policy (*scaling_policy*)
 Creates a new Scaling Policy.

Parameters scaling_policy (*boto.ec2.autoscale.policy.ScalingPolicy*)
 – ScalingPolicy object.

create_scheduled_group_action (*as_group, name, time=None, desired_capacity=None,*
min_size=None, max_size=None, start_time=None,
end_time=None, recurrence=None)

Creates a scheduled scaling action for a Auto Scaling group. If you leave a parameter unspecified, the corresponding value remains unchanged in the affected Auto Scaling group.

Parameters

- **as_group** (*string*) – The auto scaling group to get activities on.
- **name** (*string*) – Scheduled action name.
- **time** (*datetime.datetime*) – The time for this action to start. (Depracted)
- **desired_capacity** (*int*) – The number of EC2 instances that should be running in this group.
- **min_size** (*int*) – The minimum size for the new auto scaling group.
- **max_size** (*int*) – The minimum size for the new auto scaling group.
- **start_time** (*datetime.datetime*) – The time for this action to start. When Start-Time and EndTime are specified with Recurrence, they form the boundaries of when the recurring action will start and stop.
- **end_time** (*datetime.datetime*) – The time for this action to end. When Start-Time and EndTime are specified with Recurrence, they form the boundaries of when the recurring action will start and stop.
- **recurrence** (*string*) – The time when recurring future actions will start. Start time is specified by the user following the Unix cron syntax format. EXAMPLE: ‘0 10 * * *’

delete_auto_scaling_group (*name, force_delete=False*)

Deletes the specified auto scaling group if the group has no instances and no scaling activities in progress.

delete_launch_configuration (*launch_config_name*)

Deletes the specified LaunchConfiguration.

The specified launch configuration must not be attached to an Auto Scaling group. Once this call completes, the launch configuration is no longer available for use.

delete_notification_configuration (*autoscale_group, topic*)

Deletes notifications created by put_notification_configuration.

Parameters

- **autoscale_group** (*str* or *boto.ec2.autoscale.group.AutoScalingGroup* object) – The Auto Scaling group to put notification configuration on.
- **topic** (*str*) – The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.

delete_policy (*policy_name, autoscale_group=None*)

Delete a policy.

Parameters

- **policy_name** (*str*) – The name or ARN of the policy to delete.
- **autoscale_group** (*str*) – The name of the autoscale group.

delete_scheduled_action (*scheduled_action_name, autoscale_group=None*)

Deletes a previously scheduled action.

Parameters

- **scheduled_action_name** (*str*) – The name of the action you want to delete.

- **autoscale_group** (*str*) – The name of the autoscale group.

delete_tags (*tags*)

Deletes existing tags for an Auto Scaling group.

Parameters **tags** (List of *boto.ec2.autoscale.tag.Tag*) – The new or updated tags.

detach_instances (*name, instance_ids, decrement_capacity=True*)

Detach instances from an Auto Scaling group.

Parameters

- **name** (*str*) – The name of the Auto Scaling group from which to detach instances.
- **instance_ids** (*list*) – Instance ids to be detached from the Auto Scaling group.
- **decrement_capacity** (*bool*) – Whether to decrement the size of the Auto Scaling group or not.

disable_metrics_collection (*as_group, metrics=None*)

Disables monitoring of group metrics for the Auto Scaling group specified in `AutoScalingGroupName`. You can specify the list of affected metrics with the `Metrics` parameter.

enable_metrics_collection (*as_group, granularity, metrics=None*)

Enables monitoring of group metrics for the Auto Scaling group specified in `AutoScalingGroupName`. You can specify the list of enabled metrics with the `Metrics` parameter.

Auto scaling metrics collection can be turned on only if the `InstanceMonitoring.Enabled` flag, in the Auto Scaling group's launch configuration, is set to true.

Parameters

- **autoscale_group** (*string*) – The auto scaling group to get activities on.
- **granularity** (*string*) – The granularity to associate with the metrics to collect. Currently, the only legal granularity is “1Minute”.
- **metrics** (*string list*) – The list of metrics to collect. If no metrics are specified, all metrics are enabled.

execute_policy (*policy_name, as_group=None, honor_cooldown=None*)

get_account_limits ()

Returns the limits for the Auto Scaling resources currently granted for your AWS account.

get_all_activities (*autoscale_group, activity_ids=None, max_records=None, next_token=None*)

Get all activities for the given autoscaling group.

This action supports pagination by returning a token if there are more pages to retrieve. To get the next page, call this action again with the returned token as the `NextToken` parameter

Parameters

- **autoscale_group** (*str* or *boto.ec2.autoscale.group.AutoScalingGroup* object) – The auto scaling group to get activities on.
- **max_records** (*int*) – Maximum amount of activities to return.

Return type *list*

Returns List of *boto.ec2.autoscale.activity.Activity* instances.

get_all_adjustment_types ()

get_all_autoscaling_instances (*instance_ids=None, max_records=None, next_token=None*)

Returns a description of each Auto Scaling instance in the `instance_ids` list. If a list is not provided, the service returns the full details of all instances up to a maximum of fifty.

This action supports pagination by returning a token if there are more pages to retrieve. To get the next page, call this action again with the returned token as the `NextToken` parameter.

Parameters

- **instance_ids** (*list*) – List of Autoscaling Instance IDs which should be searched for.
- **max_records** (*int*) – Maximum number of results to return.

Return type `list`

Returns List of `boto.ec2.autoscale.instance.Instance` objects.

get_all_groups (*names=None, max_records=None, next_token=None*)

Returns a full description of each Auto Scaling group in the given list. This includes all Amazon EC2 instances that are members of the group. If a list of names is not provided, the service returns the full details of all Auto Scaling groups.

This action supports pagination by returning a token if there are more pages to retrieve. To get the next page, call this action again with the returned token as the `NextToken` parameter.

Parameters

- **names** (*list*) – List of group names which should be searched for.
- **max_records** (*int*) – Maximum amount of groups to return.

Return type `list`

Returns List of `boto.ec2.autoscale.group.AutoScalingGroup` instances.

get_all_launch_configurations (***kwargs*)

Returns a full description of the launch configurations given the specified names.

If no names are specified, then the full details of all launch configurations are returned.

Parameters

- **names** (*list*) – List of configuration names which should be searched for.
- **max_records** (*int*) – Maximum amount of configurations to return.
- **next_token** (*str*) – If you have more results than can be returned at once, pass in this parameter to page through all results.

Return type `list`

Returns List of `boto.ec2.autoscale.launchconfig.LaunchConfiguration` instances.

get_all_metric_collection_types ()

Returns a list of metrics and a corresponding list of granularities for each metric.

get_all_policies (*as_group=None, policy_names=None, max_records=None, next_token=None*)

Returns descriptions of what each policy does. This action supports pagination. If the response includes a token, there are more records available. To get the additional records, repeat the request with the response token as the `NextToken` parameter.

If no group name or list of policy names are provided, all available policies are returned.

Parameters

- **as_group** (*str*) – The name of the `boto.ec2.autoscale.group.AutoScalingGroup` to filter for.
- **policy_names** (*list*) – List of policy names which should be searched for.
- **max_records** (*int*) – Maximum amount of groups to return.
- **next_token** (*str*) – If you have more results than can be returned at once, pass in this parameter to page through all results.

get_all_scaling_process_types ()

Returns scaling process types for use in the ResumeProcesses and SuspendProcesses actions.

get_all_scheduled_actions (*as_group=None, start_time=None, end_time=None, scheduled_actions=None, max_records=None, next_token=None*)

get_all_tags (*filters=None, max_records=None, next_token=None*)

Lists the Auto Scaling group tags.

This action supports pagination by returning a token if there are more pages to retrieve. To get the next page, call this action again with the returned token as the NextToken parameter.

Parameters

- **filters** (*dict*) – The value of the filter type used to identify the tags to be returned. NOT IMPLEMENTED YET.
- **max_records** (*int*) – Maximum number of tags to return.

Return type *list*

Returns List of `boto.ec2.autoscale.tag.Tag` instances.

get_termination_policies ()

Gets all valid termination policies.

These values can then be used as the `termination_policies` arg when creating and updating autoscale groups.

put_notification_configuration (*autoscale_group, topic, notification_types*)

Configures an Auto Scaling group to send notifications when specified events take place.

Parameters

- **autoscale_group** (*str* or `boto.ec2.autoscale.group.AutoScalingGroup` object) – The Auto Scaling group to put notification configuration on.
- **topic** (*str*) – The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic.
- **notification_types** (*list*) – The type of events that will trigger the notification. Valid types are: ‘autoscaling:EC2_INSTANCE_LAUNCH’, ‘autoscaling:EC2_INSTANCE_LAUNCH_ERROR’, ‘autoscaling:EC2_INSTANCE_TERMINATE’, ‘autoscaling:EC2_INSTANCE_TERMINATE_ERROR’, ‘autoscaling:TEST_NOTIFICATION’

resume_processes (*as_group, scaling_processes=None*)

Resumes Auto Scaling processes for an Auto Scaling group.

Parameters

- **as_group** (*string*) – The auto scaling group to resume processes on.
- **scaling_processes** (*list*) – Processes you want to resume. If omitted, all processes will be resumed.

set_desired_capacity (*group_name*, *desired_capacity*, *honor_cooldown=False*)

Adjusts the desired size of the AutoScalingGroup by initiating scaling activities. When reducing the size of the group, it is not possible to define which Amazon EC2 instances will be terminated. This applies to any Auto Scaling decisions that might result in terminating instances.

Parameters

- **group_name** (*string*) – name of the auto scaling group
- **desired_capacity** (*integer*) – new capacity setting for auto scaling group
- **honor_cooldown** (*boolean*) – by default, overrides any cooldown period

set_instance_health (*instance_id*, *health_status*, *should_respect_grace_period=True*)

Explicitly set the health status of an instance.

Parameters

- **instance_id** (*str*) – The identifier of the EC2 instance.
- **health_status** (*str*) – The health status of the instance. “Healthy” means that the instance is healthy and should remain in service. “Unhealthy” means that the instance is unhealthy. Auto Scaling should terminate and replace it.
- **should_respect_grace_period** (*bool*) – If True, this call should respect the grace period associated with the group.

suspend_processes (*as_group*, *scaling_processes=None*)

Suspends Auto Scaling processes for an Auto Scaling group.

Parameters

- **as_group** (*string*) – The auto scaling group to suspend processes on.
- **scaling_processes** (*list*) – Processes you want to suspend. If omitted, all processes will be suspended.

terminate_instance (*instance_id*, *decrement_capacity=True*)

Terminates the specified instance. The desired group size can also be adjusted, if desired.

Parameters

- **instance_id** (*str*) – The ID of the instance to be terminated.
- **decrement_capacity** – Whether to decrement the size of the autoscaling group or not.

`boto.ec2.autoscale.connect_to_region` (*region_name*, ***kw_params*)

Given a valid region name, return a `boto.ec2.autoscale.AutoScaleConnection`.

Parameters **region_name** (*str*) – The name of the region to connect to.

Return type `boto.ec2.AutoScaleConnection` or `None`

Returns A connection to the given region, or `None` if an invalid region name is given

`boto.ec2.autoscale.regions` ()

Get all available regions for the Auto Scaling service.

Return type `list`

Returns A list of `boto.RegionInfo` instances

4.106.2 boto.ec2.autoscale.activity

```
class boto.ec2.autoscale.activity.Activity (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.106.3 boto.ec2.autoscale.group

```
class boto.ec2.autoscale.group.AutoScalingGroup (connection=None,          name=None,
                                                  launch_config=None,
                                                  availability_zones=None,
                                                  load_balancers=None,          de-
                                                  fault_cooldown=None,
                                                  health_check_type=None,
                                                  health_check_period=None,
                                                  placement_group=None,
                                                  vpc_zone_identifier=None,      de-
                                                  sired_capacity=None, min_size=None,
                                                  max_size=None,          tags=None,
                                                  termination_policies=None,      in-
                                                  stance_id=None, **kwargs)
```

Creates a new AutoScalingGroup with the specified name.

You must not have already used up your entire quota of AutoScalingGroups in order for this call to be successful. Once the creation request is completed, the AutoScalingGroup is ready to be used in other calls.

Parameters

- **name** (*str*) – Name of autoscaling group (required).
- **availability_zones** (*list*) – List of availability zones (required).
- **default_cooldown** (*int*) – Number of seconds after a Scaling Activity completes before any further scaling activities can start.
- **desired_capacity** (*int*) – The desired capacity for the group.
- **health_check_period** (*str*) – Length of time in seconds after a new EC2 instance comes into service that Auto Scaling starts checking its health.
- **health_check_type** (*str*) – The service you want the health status from, Amazon EC2 or Elastic Load Balancer.
- **launch_config** (*str* or *LaunchConfiguration*) – Name of launch configuration (required).
- **load_balancers** (*list*) – List of load balancers.
- **max_size** (*int*) – Maximum size of group (required).
- **min_size** (*int*) – Minimum size of group (required).
- **placement_group** (*str*) – Physical location of your cluster placement group created in Amazon EC2.
- **vpc_zone_identifier** (*str* or *list*) – A comma-separated string or python list of the subnet identifiers of the Virtual Private Cloud.
- **tags** (*list*) – List of :class:`boto.ec2.autoscale.tag.Tag`'s

- **termination_policies** (*list*) – A list of termination policies. Valid values are: “OldestInstance”, “NewestInstance”, “OldestLaunchConfiguration”, “ClosestToNextInstanceHour”, “Default”. If no value is specified, the “Default” value is used.
- **instance_id** (*str*) – The ID of the Amazon EC2 instance you want to use to create the Auto Scaling group.

Return type `boto.ec2.autoscale.group.AutoScalingGroup`

Returns An autoscale group.

cooldown

delete (*force_delete=False*)

Delete this auto-scaling group if no instances attached or no scaling activities in progress.

delete_notification_configuration (*topic*)

Deletes notifications created by `put_notification_configuration`.

endElement (*name, value, connection*)

get_activities (*activity_ids=None, max_records=50*)

Get all activities for this group.

put_notification_configuration (*topic, notification_types*)

Configures an Auto Scaling group to send notifications when specified events take place. Valid notification types are: ‘autoscaling:EC2_INSTANCE_LAUNCH’, ‘autoscaling:EC2_INSTANCE_LAUNCH_ERROR’, ‘autoscaling:EC2_INSTANCE_TERMINATE’, ‘autoscaling:EC2_INSTANCE_TERMINATE_ERROR’, ‘autoscaling:TEST_NOTIFICATION’

resume_processes (*scaling_processes=None*)

Resumes Auto Scaling processes for an Auto Scaling group.

set_capacity (*capacity*)

Set the desired capacity for the group.

shutdown_instances ()

Convenience method which shuts down all instances associated with this group.

startElement (*name, attrs, connection*)

suspend_processes (*scaling_processes=None*)

Suspends Auto Scaling processes for an Auto Scaling group.

update ()

Sync local changes with AutoScaling group.

class `boto.ec2.autoscale.group.AutoScalingGroupMetric` (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.autoscale.group.EnabledMetric` (*connection=None, metric=None, granularity=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.autoscale.group.ProcessType` (*connection=None*)

endElement (*name, value, connection*)

```
    startElement (name, attrs, connection)
```

```
class boto.ec2.autoscale.group.SuspendedProcess (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.ec2.autoscale.group.TerminationPolicies
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.106.4 boto.ec2.autoscale.instance

```
class boto.ec2.autoscale.instance.Instance (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.106.5 boto.ec2.autoscale.launchconfig

```
class boto.ec2.autoscale.launchconfig.BlockDeviceMapping (connection=None,  
                                                           device_name=None,  
                                                           virtual_name=None,  
                                                           ebs=None,  
                                                           no_device=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.ec2.autoscale.launchconfig.Ebs (connection=None, snapshot_id=None, vol-  
                                           ume_size=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.ec2.autoscale.launchconfig.InstanceMonitoring (connection=None, en-  
                                                           abled='false')
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```

class boto.ec2.autoscale.launchconfig.LaunchConfiguration (connection=None,
                                                            name=None,      im-
                                                            age_id=None,
                                                            key_name=None,
                                                            security_groups=None,
                                                            user_data=None,  in-
                                                            stance_type='m1.small',
                                                            kernel_id=None,
                                                            ramdisk_id=None,
                                                            block_device_mappings=None,
                                                            in-
                                                            stance_monitoring=False,
                                                            spot_price=None,  in-
                                                            stance_profile_name=None,
                                                            ebs_optimized=False,
                                                            asso-
                                                            ciate_public_ip_address=None,
                                                            volume_type=None,
                                                            delete_on_termination=True,
                                                            iops=None,
                                                            use_block_device_types=False,
                                                            clas-
                                                            sic_link_vpc_id=None,
                                                            clas-
                                                            sic_link_vpc_security_groups=None)

```

A launch configuration.

Parameters

- **name** (*str*) – Name of the launch configuration to create.
- **image_id** (*str*) – Unique ID of the Amazon Machine Image (AMI) which was assigned during registration.
- **key_name** (*str*) – The name of the EC2 key pair.
- **security_groups** (*list*) – Names or security group id's of the security groups with which to associate the EC2 instances or VPC instances, respectively.
- **user_data** (*str*) – The user data available to launched EC2 instances.
- **instance_type** (*str*) – The instance type
- **kernel_id** (*str*) – Kernel id for instance
- **ramdisk_id** (*str*) – RAM disk id for instance
- **block_device_mappings** (*list*) – Specifies how block devices are exposed for instances
- **instance_monitoring** (*bool*) – Whether instances in group are launched with detailed monitoring.
- **spot_price** (*float*) – The spot price you are bidding. Only applies if you are building an autoscaling group with spot instances.
- **instance_profile_name** (*string*) – The name or the Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance.
- **ebs_optimized** (*bool*) – Specifies whether the instance is optimized for EBS I/O (true) or not (false).

- **associate_public_ip_address** (*bool*) – Used for Auto Scaling groups that launch instances into an Amazon Virtual Private Cloud. Specifies whether to assign a public IP address to each instance launched in a Amazon VPC.
- **classic_link_vpc_id** (*str*) – ID of ClassicLink enabled VPC.
- **classic_link_vpc_security_groups** (*list*) – Security group id's of the security groups with which to associate the ClassicLink VPC instances.

delete ()

Delete this launch configuration.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.106.6 boto.ec2.autoscale.policy

class boto.ec2.autoscale.policy.**AdjustmentType** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.autoscale.policy.**Alarm** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.autoscale.policy.**MetricCollectionTypes** (*connection=None*)

class **BaseType** (*connection*)

arg = ''

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class **Granularity** (*connection*)

arg = 'Granularity'

class **Metric** (*connection*)

arg = 'Metric'

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.autoscale.policy.**ScalingPolicy** (*connection=None, **kwargs*)

Scaling Policy

Parameters

- **name** (*str*) – Name of scaling policy.

- **adjustment_type** (*str*) – Specifies the type of adjustment. Valid values are *ChangeInCapacity*, *ExactCapacity* and *PercentChangeInCapacity*.
- **as_name** (*str* or *int*) – Name or ARN of the Auto Scaling Group.
- **scaling_adjustment** (*int*) – Value of adjustment (type specified in *adjustment_type*).
- **min_adjustment_step** (*int*) – Value of min adjustment step required to apply the scaling policy (only make sense when use *PercentChangeInCapacity* as *adjustment_type*).
- **cooldown** (*int*) – Time (in seconds) before Alarm related Scaling Activities can start after the previous Scaling Activity ends.

```
delete ()
```

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

```
class boto.ec2.autoscale.policy.TerminationPolicies (connection=None, **kwargs)
```

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

4.106.7 boto.ec2.autoscale.request

```
class boto.ec2.autoscale.request.Request (connection=None)
```

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

4.106.8 boto.ec2.autoscale.scheduled

```
class boto.ec2.autoscale.scheduled.ScheduledUpdateGroupAction (connection=None)
```

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

4.106.9 boto.ec2.autoscale.tag

```
class boto.ec2.autoscale.tag.Tag (connection=None, key=None, value=None, propagate_at_launch=False, resource_id=None, resource_type='auto-scaling-group')
```

A name/value tag on an AutoScalingGroup resource.

Variables

- **key** – The key of the tag.
- **value** – The value of the tag.
- **propagate_at_launch** – Boolean value which specifies whether the new tag will be applied to instances launched after the tag is created.
- **resource_id** – The name of the autoscaling group.

- **resource_type** – The only supported resource type at this time is “auto-scaling-group”.

build_params (*params, i*)

Populates a dictionary with the name/value pairs necessary to identify this Tag in a request.

delete ()

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.107 AWS Lambda

4.107.1 boto.awslambda

`boto.awslambda.connect_to_region` (*region_name, **kw_params*)

`boto.awslambda.regions` ()

Get all available regions for the AWS Lambda service. :rtype: list :return: A list of `boto.regioninfo.RegionInfo`

4.107.2 boto.awslambda.layer1

class `boto.awslambda.layer1.AWSLambdaConnection` (***kwargs*)

AWS Lambda Overview

This is the AWS Lambda API Reference. The AWS Lambda Developer Guide provides additional information. For the service overview, go to ‘[What is AWS Lambda](#)’, and for information about how the service works, go to ‘[AWS LambdaL How it Works](#)’ in the AWS Lambda Developer Guide.

APIVersion = '2014-11-11'

DefaultRegionEndpoint = 'lambda.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

add_event_source (*event_source, function_name, role, batch_size=None, parameters=None*)

Identifies an Amazon Kinesis stream as the event source for an AWS Lambda function. AWS Lambda invokes the specified function when records are posted to the stream.

This is the pull model, where AWS Lambda invokes the function. For more information, go to ‘[AWS LambdaL How it Works](#)’ in the AWS Lambda Developer Guide.

This association between an Amazon Kinesis stream and an AWS Lambda function is called the event source mapping. You provide the configuration information (for example, which stream to read from and which AWS Lambda function to invoke) for the event source mapping in the request body.

This operation requires permission for the `iam:PassRole` action for the IAM role. It also requires permission for the `lambda:AddEventSource` action.

Parameters

- **event_source** (*string*) – The Amazon Resource Name (ARN) of the Amazon Kinesis stream that is the event source. Any record added to this stream causes AWS Lambda to invoke your Lambda function. AWS Lambda POSTs the Amazon Kinesis event, containing records, to your Lambda function as JSON.

- **function_name** (*string*) – The Lambda function to invoke when AWS Lambda detects an event on the stream.
- **role** (*string*) – The ARN of the IAM role (invocation role) that AWS Lambda can assume to read from the stream and invoke the function.
- **batch_size** (*integer*) – The largest number of records that AWS Lambda will give to your function in a single event. The default is 100 records.
- **parameters** (*map*) – A map (key-value pairs) defining the configuration for AWS Lambda to use when reading the event source. Currently, AWS Lambda supports only the *InitialPositionInStream* key. The valid values are: “TRIM_HORIZON” and “LATEST”. The default value is “TRIM_HORIZON”. For more information, go to ‘[ShardIteratorType](#)’ in the Amazon Kinesis Service API Reference.

delete_function (*function_name*)

Deletes the specified Lambda function code and configuration.

This operation requires permission for the *lambda:DeleteFunction* action.

Parameters **function_name** (*string*) – The Lambda function to delete.

get_event_source (*uuid*)

Returns configuration information for the specified event source mapping (see *AddEventSource*).

This operation requires permission for the *lambda:GetEventSource* action.

Parameters **uuid** (*string*) – The AWS Lambda assigned ID of the event source mapping.

get_function (*function_name*)

Returns the configuration information of the Lambda function and a presigned URL link to the .zip file you uploaded with *UploadFunction* so you can download the .zip file. Note that the URL is valid for up to 10 minutes. The configuration information is the same information you provided as parameters when uploading the function.

This operation requires permission for the *lambda:GetFunction* action.

Parameters **function_name** (*string*) – The Lambda function name.

get_function_configuration (*function_name*)

Returns the configuration information of the Lambda function. This the same information you provided as parameters when uploading the function by using *UploadFunction*.

This operation requires permission for the *lambda:GetFunctionConfiguration* operation.

Parameters **function_name** (*string*) – The name of the Lambda function for which you want to retrieve the configuration information.

invoke_async (*function_name, invoke_args*)

Submits an invocation request to AWS Lambda. Upon receiving the request, Lambda executes the specified function asynchronously. To see the logs generated by the Lambda function execution, see the CloudWatch logs console.

This operation requires permission for the *lambda:InvokeAsync* action.

Parameters

- **function_name** (*string*) – The Lambda function name.
- **invoke_args** (*blob*) – JSON that you want to provide to your Lambda function as input.

list_event_sources (*event_source_arn=None, function_name=None, marker=None, max_items=None*)

Returns a list of event source mappings. For each mapping, the API returns configuration information (see `AddEventSource`). You can optionally specify filters to retrieve specific event source mappings.

This operation requires permission for the `lambda:ListEventSources` action.

Parameters

- **event_source_arn** (*string*) – The Amazon Resource Name (ARN) of the Amazon Kinesis stream.
- **function_name** (*string*) – The name of the AWS Lambda function.
- **marker** (*string*) – Optional string. An opaque pagination token returned from a previous `ListEventSources` operation. If present, specifies to continue the list from where the returning call left off.
- **max_items** (*integer*) – Optional integer. Specifies the maximum number of event sources to return in response. This value must be greater than 0.

list_functions (*marker=None, max_items=None*)

Returns a list of your Lambda functions. For each function, the response includes the function configuration information. You must use `GetFunction` to retrieve the code for your function.

This operation requires permission for the `lambda:ListFunctions` action.

Parameters

- **marker** (*string*) – Optional string. An opaque pagination token returned from a previous `ListFunctions` operation. If present, indicates where to continue the listing.
- **max_items** (*integer*) – Optional integer. Specifies the maximum number of AWS Lambda functions to return in response. This parameter value must be greater than 0.

make_request (*verb, resource, headers=None, data="", expected_status=None, params=None*)

Makes a request to the server, with stock multiple-retry logic.

remove_event_source (*uuid*)

Removes an event source mapping. This means AWS Lambda will no longer invoke the function for events in the associated source.

This operation requires permission for the `lambda:RemoveEventSource` action.

Parameters **uuid** (*string*) – The event source mapping ID.

update_function_configuration (*function_name, role=None, handler=None, description=None, timeout=None, memory_size=None*)

Updates the configuration parameters for the specified Lambda function by using the values provided in the request. You provide only the parameters you want to change. This operation must only be used on an existing Lambda function and cannot be used to update the function's code.

This operation requires permission for the `lambda:UpdateFunctionConfiguration` action.

Parameters

- **function_name** (*string*) – The name of the Lambda function.
- **role** (*string*) – The Amazon Resource Name (ARN) of the IAM role that Lambda will assume when it executes your function.
- **handler** (*string*) – The function that Lambda calls to begin executing your function. For Node.js, it is the `module-name.export` value in your function.

- **description** (*string*) – A short user-defined function description. Lambda does not use this value. Assign a meaningful description as you see fit.
- **timeout** (*integer*) – The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- **memory_size** (*integer*) – The amount of memory, in MB, your Lambda function is given. Lambda uses this memory size to infer the amount of CPU allocated to your function. Your function use-case determines your CPU and memory requirements. For example, a database operation might need less memory compared to an image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.

upload_function (*function_name, function_zip, runtime, role, handler, mode, description=None, timeout=None, memory_size=None*)

Creates a new Lambda function or updates an existing function. The function metadata is created from the request parameters, and the code for the function is provided by a .zip file in the request body. If the function name already exists, the existing Lambda function is updated with the new code and metadata.

This operation requires permission for the *lambda:UploadFunction* action.

Parameters

- **function_name** (*string*) – The name you want to assign to the function you are uploading. The function names appear in the console and are returned in the ListFunctions API. Function names are used to specify functions to other AWS Lambda APIs, such as InvokeAsync.
- **function_zip** (*blob*) – A .zip file containing your packaged source code. For more information about creating a .zip file, go to [‘AWS LambdaL How it Works’](#) in the AWS Lambda Developer Guide.
- **runtime** (*string*) – The runtime environment for the Lambda function you are uploading. Currently, Lambda supports only “nodejs” as the runtime.
- **role** (*string*) – The Amazon Resource Name (ARN) of the IAM role that Lambda assumes when it executes your function to access any other Amazon Web Services (AWS) resources.
- **handler** (*string*) – The function that Lambda calls to begin execution. For Node.js, it is the module-name . export value in your function.
- **mode** (*string*) – How the Lambda function will be invoked. Lambda supports only the “event” mode.
- **description** (*string*) – A short, user-defined function description. Lambda does not use this value. Assign a meaningful description as you see fit.
- **timeout** (*integer*) – The function execution time at which Lambda should terminate the function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.
- **memory_size** (*integer*) – The amount of memory, in MB, your Lambda function is given. Lambda uses this memory size to infer the amount of CPU allocated to your function. Your function use-case determines your CPU and memory requirements. For example, database operation might need less memory compared to image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.

4.107.3 boto.awslambda.exceptions

```
exception boto.awslambda.exceptions.InvalidParameterValueException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.awslambda.exceptions.InvalidRequestContentException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.awslambda.exceptions.ResourceNotFoundException (status,    reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.awslambda.exceptions.ServiceException (status,    reason,    body=None,
                                                                    *args)
```

4.108 Elastic Beanstalk

4.108.1 boto.beanstalk

```
boto.beanstalk.connect_to_region (region_name, **kw_params)
```

```
boto.beanstalk.regions ()
```

Get all available regions for the AWS Elastic Beanstalk service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.108.2 boto.beanstalk.layer1

```
class boto.beanstalk.layer1.Layer1 (aws_access_key_id=None, aws_secret_access_key=None,
                                     is_secure=True,          port=None,          proxy=None,
                                     proxy_port=None, proxy_user=None, proxy_pass=None,
                                     debug=0, https_connection_factory=None, region=None,
                                     path='/',  api_version=None, security_token=None,
                                     profile_name=None)
```

```
APIVersion = '2010-12-01'
```

```
DefaultRegionEndpoint = 'elasticbeanstalk.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
check_dns_availability (cname_prefix)
```

Checks if the specified CNAME is available.

Parameters `cname_prefix` (*string*) – The prefix used when this CNAME is reserved.

```
create_application (application_name, description=None)
```

Creates an application that has one configuration template named default and no application versions.

Parameters

- **application_name** (*string*) – The name of the application. Constraint: This name must be unique within your account. If the specified name already exists, the action returns an `InvalidParameterValue` error.
- **description** (*string*) – Describes the application.

Raises `TooManyApplicationsException`

create_application_version (*application_name*, *version_label*, *description=None*, *s3_bucket=None*, *s3_key=None*, *auto_create_application=None*)

Creates an application version for the specified application.

Parameters

- **application_name** (*string*) – The name of the application. If no application is found with this name, and `AutoCreateApplication` is false, returns an `InvalidParameterValue` error.
- **version_label** (*string*) – A label identifying this version. Constraint: Must be unique per application. If an application version already exists with this label for the specified application, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **description** (*string*) – Describes this version.
- **s3_bucket** (*string*) – The Amazon S3 bucket where the data is located.
- **s3_key** (*string*) – The Amazon S3 key where the data is located. Both `s3_bucket` and `s3_key` must be specified in order to use a specific source bundle. If both of these values are not specified the sample application will be used.
- **auto_create_application** (*boolean*) – Determines how the system behaves if the specified application for this version does not already exist: `true`: Automatically creates the specified application for this version if it does not already exist. `false`: Returns an `InvalidParameterValue` if the specified application for this version does not already exist. Default: `false` Valid Values: `true` | `false`

Raises `TooManyApplicationsException`, `TooManyApplicationVersionsException`, `InsufficientPrivilegesException`, `S3LocationNotInServiceRegionException`

create_configuration_template (*application_name*, *template_name*, *solution_stack_name=None*, *source_configuration_application_name=None*, *source_configuration_template_name=None*, *environment_id=None*, *description=None*, *option_settings=None*)

Creates a configuration template.

Templates are associated with a specific application and are used to deploy different versions of the application with the same configuration settings.

Parameters

- **application_name** (*string*) – The name of the application to associate with this configuration template. If no application is found with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **template_name** (*string*) – The name of the configuration template. Constraint: This name must be unique per application. Default: If a configuration template already exists with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error.
- **solution_stack_name** (*string*) – The name of the solution stack used by this configuration. The solution stack specifies the operating system, architecture, and application server for a configuration template. It determines the set of configuration options

as well as the possible and default values. Use `ListAvailableSolutionStacks` to obtain a list of available solution stacks. Default: If the `SolutionStackName` is not specified and the source configuration parameter is blank, AWS Elastic Beanstalk uses the default solution stack. If not specified and the source configuration parameter is specified, AWS Elastic Beanstalk uses the same solution stack as the source configuration template.

- **source_configuration_application_name** (*string*) – The name of the application associated with the configuration.
- **source_configuration_template_name** (*string*) – The name of the configuration template.
- **environment_id** (*string*) – The ID of the environment used with this configuration template.
- **description** (*string*) – Describes this configuration.
- **option_settings** (*list*) – If specified, AWS Elastic Beanstalk sets the specified configuration option to the requested value. The new value overrides the value obtained from the solution stack or the source configuration template.

Raises `InsufficientPrivilegesException`, `TooManyConfigurationTemplatesException`

create_environment (*application_name*, *environment_name*, *version_label=None*, *template_name=None*, *solution_stack_name=None*, *cname_prefix=None*, *description=None*, *option_settings=None*, *options_to_remove=None*, *tier_name=None*, *tier_type=None*, *tier_version='1.0'*)

Launches an environment for the application using a configuration.

Parameters

- **application_name** (*string*) – The name of the application that contains the version to be deployed. If no application is found with this name, `CreateEnvironment` returns an `InvalidParameterValue` error.
- **environment_name** (*string*) – A unique name for the deployment environment. Used in the application URL. Constraint: Must be from 4 to 23 characters in length. The name can contain only letters, numbers, and hyphens. It cannot start or end with a hyphen. This name must be unique in your account. If the specified name already exists, AWS Elastic Beanstalk returns an `InvalidParameterValue` error. Default: If the `CNAME` parameter is not specified, the environment name becomes part of the `CNAME`, and therefore part of the visible URL for your application.
- **version_label** (*string*) – The name of the application version to deploy. If the specified application has no associated application versions, AWS Elastic Beanstalk `UpdateEnvironment` returns an `InvalidParameterValue` error. Default: If not specified, AWS Elastic Beanstalk attempts to launch the most recently created application version.
- **template_name** (*string*) – The name of the configuration template to use in deployment. If no configuration template is found with this name, AWS Elastic Beanstalk returns an `InvalidParameterValue` error. Condition: You must specify either this parameter or a `SolutionStackName`, but not both. If you specify both, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error. If you do not specify either, AWS Elastic Beanstalk returns a `MissingRequiredParameter` error.
- **solution_stack_name** (*string*) – This is an alternative to specifying a configuration name. If specified, AWS Elastic Beanstalk sets the configuration values to the default values associated with the specified solution stack. Condition: You must specify either this or a `TemplateName`, but not both. If you specify both, AWS Elastic Beanstalk returns an `InvalidParameterCombination` error. If you do not specify either, AWS Elastic Beanstalk returns a `MissingRequiredParameter` error.

- **cname_prefix** (*string*) – If specified, the environment attempts to use this value as the prefix for the CNAME. If not specified, the environment uses the environment name.
- **description** (*string*) – Describes this environment.
- **option_settings** (*list*) – If specified, AWS Elastic Beanstalk sets the specified configuration options to the requested value in the configuration set for the new environment. These override the values obtained from the solution stack or the configuration template. Each element in the list is a tuple of (Namespace, OptionName, Value), for example:

```
[('aws:autoscaling:launchconfiguration',
  'Ec2KeyName', 'mykeypair')]
```

- **options_to_remove** (*list*) – A list of custom user-defined configuration options to remove from the configuration set for this new environment.
- **tier_name** (*string*) – The name of the tier. Valid values are “WebServer” and “Worker”. Defaults to “WebServer”. The `tier_name` and a `tier_type` parameters are related and the values provided must be valid. The possible combinations are:
 - “WebServer” and “Standard” (the default)
 - “Worker” and “SQS/HTTP”
- **tier_type** (*string*) – The type of the tier. Valid values are “Standard” if `tier_name` is “WebServer” and “SQS/HTTP” if `tier_name` is “Worker”. Defaults to “Standard”.

Raises TooManyEnvironmentsException, InsufficientPrivilegesException

create_storage_location ()

Creates the Amazon S3 storage location for the account. This location is used to store user log files.

Raises TooManyBucketsException, S3SubscriptionRequiredException, InsufficientPrivilegesException

delete_application (*application_name*, *terminate_env_by_force=None*)

Deletes the specified application along with all associated versions and configurations. The application versions will not be deleted from your Amazon S3 bucket.

Parameters

- **application_name** (*string*) – The name of the application to delete.
- **terminate_env_by_force** (*boolean*) – When set to true, running environments will be terminated before deleting the application.

Raises OperationInProgressException

delete_application_version (*application_name*, *version_label*, *delete_source_bundle=None*)

Deletes the specified version from the specified application.

Parameters

- **application_name** (*string*) – The name of the application to delete releases from.
- **version_label** (*string*) – The label of the version to delete.
- **delete_source_bundle** (*boolean*) – Indicates whether to delete the associated source bundle from Amazon S3. Valid Values: true | false

Raises SourceBundleDeletionException, InsufficientPrivilegesException, OperationInProgressException, S3LocationNotInServiceRegionException

delete_configuration_template (*application_name*, *template_name*)

Deletes the specified configuration template.

Parameters

- **application_name** (*string*) – The name of the application to delete the configuration template from.
- **template_name** (*string*) – The name of the configuration template to delete.

Raises OperationInProgressException

delete_environment_configuration (*application_name*, *environment_name*)

Deletes the draft configuration associated with the running environment. Updating a running environment with any configuration changes creates a draft configuration set. You can get the draft configuration using DescribeConfigurationSettings while the update is in progress or if the update fails. The DeploymentStatus for the draft configuration indicates whether the deployment is in process or has failed. The draft configuration remains in existence until it is deleted with this action.

Parameters

- **application_name** (*string*) – The name of the application the environment is associated with.
- **environment_name** (*string*) – The name of the environment to delete the draft configuration from.

describe_application_versions (*application_name=None*, *version_labels=None*)

Returns descriptions for existing application versions.

Parameters

- **application_name** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to only include ones that are associated with the specified application.
- **version_labels** (*list*) – If specified, restricts the returned descriptions to only include ones that have the specified version labels.

describe_applications (*application_names=None*)

Returns the descriptions of existing applications.

Parameters **application_names** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to only include those with the specified names.

describe_configuration_options (*application_name=None*, *template_name=None*, *environment_name=None*, *solution_stack_name=None*, *options=None*)

Describes configuration options used in a template or environment.

Describes the configuration options that are used in a particular configuration template or environment, or that a specified solution stack defines. The description includes the values the options, their default values, and an indication of the required action on a running environment if an option value is changed.

Parameters

- **application_name** (*string*) – The name of the application associated with the configuration template or environment. Only needed if you want to describe the configuration options associated with either the configuration template or environment.
- **template_name** (*string*) – The name of the configuration template whose configuration options you want to describe.
- **environment_name** (*string*) – The name of the environment whose configuration options you want to describe.

- **solution_stack_name** (*string*) – The name of the solution stack whose configuration options you want to describe.
- **options** (*list*) – If specified, restricts the descriptions to only the specified options.

describe_configuration_settings (*application_name*, *template_name=None*, *environment_name=None*)

Returns a description of the settings for the specified configuration set, that is, either a configuration template or the configuration set associated with a running environment. When describing the settings for the configuration set associated with a running environment, it is possible to receive two sets of setting descriptions. One is the deployed configuration set, and the other is a draft configuration of an environment that is either in the process of deployment or that failed to deploy.

Parameters

- **application_name** (*string*) – The application for the environment or configuration template.
- **template_name** (*string*) – The name of the configuration template to describe. Conditional: You must specify either this parameter or an EnvironmentName, but not both. If you specify both, AWS Elastic Beanstalk returns an InvalidParameterCombination error. If you do not specify either, AWS Elastic Beanstalk returns a MissingRequiredParameter error.
- **environment_name** (*string*) – The name of the environment to describe. Condition: You must specify either this or a TemplateName, but not both. If you specify both, AWS Elastic Beanstalk returns an InvalidParameterCombination error. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.

describe_environment_resources (*environment_id=None*, *environment_name=None*)

Returns AWS resources for this environment.

Parameters

- **environment_id** (*string*) – The ID of the environment to retrieve AWS resource usage data. Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **environment_name** (*string*) – The name of the environment to retrieve AWS resource usage data. Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.

Raises InsufficientPrivilegesException

describe_environments (*application_name=None*, *version_label=None*, *environment_ids=None*, *environment_names=None*, *include_deleted=None*, *included_deleted_back_to=None*)

Returns descriptions for existing environments.

Parameters

- **application_name** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that are associated with this application.
- **version_label** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that are associated with this application version.
- **environment_ids** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that have the specified IDs.
- **environment_names** (*list*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those that have the specified names.

- **include_deleted** (*boolean*) – Indicates whether to include deleted environments: true: Environments that have been deleted after IncludedDeletedBackTo are displayed. false: Do not include deleted environments.
- **included_deleted_back_to** (*timestamp*) – If specified when IncludeDeleted is set to true, then environments deleted after this date are displayed.

describe_events (*application_name=None, version_label=None, template_name=None, environment_id=None, environment_name=None, request_id=None, severity=None, start_time=None, end_time=None, max_records=None, next_token=None*)

Returns event descriptions matching criteria up to the last 6 weeks.

Parameters

- **application_name** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to include only those associated with this application.
- **version_label** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this application version.
- **template_name** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that are associated with this environment configuration.
- **environment_id** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **environment_name** (*string*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those associated with this environment.
- **request_id** (*string*) – If specified, AWS Elastic Beanstalk restricts the described events to include only those associated with this request ID.
- **severity** (*string*) – If specified, limits the events returned from this call to include only those with the specified severity or higher.
- **start_time** (*timestamp*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur on or after this time.
- **end_time** (*timestamp*) – If specified, AWS Elastic Beanstalk restricts the returned descriptions to those that occur up to, but not including, the EndTime.
- **max_records** (*integer*) – Specifies the maximum number of events that can be returned, beginning with the most recent event.
- **next_token** (*string*) – Pagination token. If specified, the events return the next batch of results.

list_available_solution_stacks ()

Returns a list of the available solution stack names.

rebuild_environment (*environment_id=None, environment_name=None*)

Deletes and recreates all of the AWS resources (for example: the Auto Scaling group, load balancer, etc.) for a specified environment and forces a restart.

Parameters

- **environment_id** (*string*) – The ID of the environment to rebuild. Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **environment_name** (*string*) – The name of the environment to rebuild. Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.

Raises `InvalidParameterValue` – If `environment_name` doesn't refer to a currently active environment

Raises `InsufficientPrivilegesException`

`request_environment_info` (*info_type='tail', environment_id=None, environment_name=None*)

Initiates a request to compile the specified type of information of the deployed environment. Setting the `InfoType` to `tail` compiles the last lines from the application server log files of every Amazon EC2 instance in your environment. Use `RetrieveEnvironmentInfo` to access the compiled information.

Parameters

- **`info_type`** (*string*) – The type of information to request.
- **`environment_id`** (*string*) – The ID of the environment of the requested data. If no such environment is found, `RequestEnvironmentInfo` returns an `InvalidParameterValue` error. Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **`environment_name`** (*string*) – The name of the environment of the requested data. If no such environment is found, `RequestEnvironmentInfo` returns an `InvalidParameterValue` error. Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

`restart_app_server` (*environment_id=None, environment_name=None*)

Causes the environment to restart the application container server running on each Amazon EC2 instance.

Parameters

- **`environment_id`** (*string*) – The ID of the environment to restart the server for. Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **`environment_name`** (*string*) – The name of the environment to restart the server for. Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

`retrieve_environment_info` (*info_type='tail', environment_id=None, environment_name=None*)

Retrieves the compiled information from a `RequestEnvironmentInfo` request.

Parameters

- **`info_type`** (*string*) – The type of information to retrieve.
- **`environment_id`** (*string*) – The ID of the data's environment. If no such environment is found, returns an `InvalidParameterValue` error. Condition: You must specify either this or an `EnvironmentName`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.
- **`environment_name`** (*string*) – The name of the data's environment. If no such environment is found, returns an `InvalidParameterValue` error. Condition: You must specify either this or an `EnvironmentId`, or both. If you do not specify either, AWS Elastic Beanstalk returns `MissingRequiredParameter` error.

`swap_environment_cnames` (*source_environment_id=None, source_environment_name=None, destination_environment_id=None, destination_environment_name=None*)

Swaps the CNAMEs of two environments.

Parameters

- **source_environment_id** (*string*) – The ID of the source environment. Condition: You must specify at least the SourceEnvironmentID or the SourceEnvironmentName. You may also specify both. If you specify the SourceEnvironmentId, you must specify the DestinationEnvironmentId.
- **source_environment_name** (*string*) – The name of the source environment. Condition: You must specify at least the SourceEnvironmentID or the SourceEnvironmentName. You may also specify both. If you specify the SourceEnvironmentName, you must specify the DestinationEnvironmentName.
- **destination_environment_id** (*string*) – The ID of the destination environment. Condition: You must specify at least the DestinationEnvironmentID or the DestinationEnvironmentName. You may also specify both. You must specify the SourceEnvironmentId with the DestinationEnvironmentId.
- **destination_environment_name** (*string*) – The name of the destination environment. Condition: You must specify at least the DestinationEnvironmentID or the DestinationEnvironmentName. You may also specify both. You must specify the SourceEnvironmentName with the DestinationEnvironmentName.

terminate_environment (*environment_id=None, environment_name=None, terminate_resources=None*)

Terminates the specified environment.

Parameters

- **environment_id** (*string*) – The ID of the environment to terminate. Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **environment_name** (*string*) – The name of the environment to terminate. Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **terminate_resources** (*boolean*) – Indicates whether the associated AWS resources should shut down when the environment is terminated: true: (default) The user AWS resources (for example, the Auto Scaling group, LoadBalancer, etc.) are terminated along with the environment. false: The environment is removed from the AWS Elastic Beanstalk but the AWS resources continue to operate. For more information, see the AWS Elastic Beanstalk User Guide. Default: true Valid Values: true | false

Raises InsufficientPrivilegesException

update_application (*application_name, description=None*)

Updates the specified application to have the specified properties.

Parameters

- **application_name** (*string*) – The name of the application to update. If no such application is found, UpdateApplication returns an InvalidParameterValue error.
- **description** (*string*) – A new description for the application. Default: If not specified, AWS Elastic Beanstalk does not update the description.

update_application_version (*application_name, version_label, description=None*)

Updates the application version to have the properties.

Parameters

- **application_name** (*string*) – The name of the application associated with this version. If no application is found with this name, UpdateApplication returns an InvalidParameterValue error.

- **version_label** (*string*) – The name of the version to update. If no application version is found with this label, UpdateApplication returns an InvalidParameterValue error.
- **description** (*string*) – A new description for this release.

update_configuration_template (*application_name, template_name, description=None, option_settings=None, options_to_remove=None*)

Updates the specified configuration template to have the specified properties or configuration option values.

Parameters

- **application_name** (*string*) – The name of the application associated with the configuration template to update. If no application is found with this name, UpdateConfigurationTemplate returns an InvalidParameterValue error.
- **template_name** (*string*) – The name of the configuration template to update. If no configuration template is found with this name, UpdateConfigurationTemplate returns an InvalidParameterValue error.
- **description** (*string*) – A new description for the configuration.
- **option_settings** (*list*) – A list of configuration option settings to update with the new specified option value.
- **options_to_remove** (*list*) – A list of configuration options to remove from the configuration set. Constraint: You can remove only UserDefined configuration options.

Raises InsufficientPrivilegesException

update_environment (*environment_id=None, environment_name=None, version_label=None, template_name=None, description=None, option_settings=None, options_to_remove=None, tier_name=None, tier_type=None, tier_version='1.0'*)

Updates the environment description, deploys a new application version, updates the configuration settings to an entirely new configuration template, or updates select configuration option values in the running environment. Attempting to update both the release and configuration is not allowed and AWS Elastic Beanstalk returns an InvalidParameterCombination error. When updating the configuration settings to a new template or individual settings, a draft configuration is created and DescribeConfigurationSettings for this environment returns two setting descriptions with different DeploymentStatus values.

Parameters

- **environment_id** (*string*) – The ID of the environment to update. If no environment with this ID exists, AWS Elastic Beanstalk returns an InvalidParameterValue error. Condition: You must specify either this or an EnvironmentName, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **environment_name** (*string*) – The name of the environment to update. If no environment with this name exists, AWS Elastic Beanstalk returns an InvalidParameterValue error. Condition: You must specify either this or an EnvironmentId, or both. If you do not specify either, AWS Elastic Beanstalk returns MissingRequiredParameter error.
- **version_label** (*string*) – If this parameter is specified, AWS Elastic Beanstalk deploys the named application version to the environment. If no such application version is found, returns an InvalidParameterValue error.
- **template_name** (*string*) – If this parameter is specified, AWS Elastic Beanstalk deploys this configuration template to the environment. If no such configuration template is found, AWS Elastic Beanstalk returns an InvalidParameterValue error.
- **description** (*string*) – If this parameter is specified, AWS Elastic Beanstalk updates the description of this environment.

- **option_settings** (*list*) – If specified, AWS Elastic Beanstalk updates the configuration set associated with the running environment and sets the specified configuration options to the requested value.
- **options_to_remove** (*list*) – A list of custom user-defined configuration options to remove from the configuration set for this environment.
- **tier_name** (*string*) – The name of the tier. Valid values are “WebServer” and “Worker”. Defaults to “WebServer”. The `tier_name` and a `tier_type` parameters are related and the values provided must be valid. The possible combinations are:
 - “WebServer” and “Standard” (the default)
 - “Worker” and “SQS/HTTP”
- **tier_type** (*string*) – The type of the tier. Valid values are “Standard” if `tier_name` is “WebServer” and “SQS/HTTP” if `tier_name` is “Worker”. Defaults to “Standard”.

Raises `InsufficientPrivilegesException`

validate_configuration_settings (*application_name*, *option_settings*, *template_name=None*, *environment_name=None*)

Takes a set of configuration settings and either a configuration template or environment, and determines whether those values are valid. This action returns a list of messages indicating any errors or warnings associated with the selection of option values.

Parameters

- **application_name** (*string*) – The name of the application that the configuration template or environment belongs to.
- **template_name** (*string*) – The name of the configuration template to validate the settings against. Condition: You cannot specify both this and an environment name.
- **environment_name** (*string*) – The name of the environment to validate the settings against. Condition: You cannot specify both this and a configuration template name.
- **option_settings** (*list*) – A list of the options and desired values to evaluate.

Raises `InsufficientPrivilegesException`

4.108.3 boto.beanstalk.response

Classify responses from layer1 and strict type values.

```
class boto.beanstalk.response.ApplicationDescription (response)
```

```
class boto.beanstalk.response.ApplicationVersionDescription (response)
```

```
class boto.beanstalk.response.AutoScalingGroup (response)
```

```
class boto.beanstalk.response.BaseObject
```

```
class boto.beanstalk.response.CheckDNSAvailabilityResponse (response)
```

```
class boto.beanstalk.response.CheckDnsAvailabilityResponse (response)
```

```
class boto.beanstalk.response.ConfigurationOptionDescription (response)
```

```
class boto.beanstalk.response.ConfigurationOptionSetting (response)
```

```
class boto.beanstalk.response.ConfigurationSettingsDescription (response)
```

```
class boto.beanstalk.response.CreateApplicationResponse (response)
```

```
class boto.beanstalk.response.CreateApplicationVersionResponse (response)
class boto.beanstalk.response.CreateConfigurationTemplateResponse (response)
class boto.beanstalk.response.CreateEnvironmentResponse (response)
class boto.beanstalk.response.CreateStorageLocationResponse (response)
class boto.beanstalk.response.DeleteApplicationResponse (response)
class boto.beanstalk.response.DeleteApplicationVersionResponse (response)
class boto.beanstalk.response.DeleteConfigurationTemplateResponse (response)
class boto.beanstalk.response.DeleteEnvironmentConfigurationResponse (response)
class boto.beanstalk.response.DescribeApplicationVersionsResponse (response)
class boto.beanstalk.response.DescribeApplicationsResponse (response)
class boto.beanstalk.response.DescribeConfigurationOptionsResponse (response)
class boto.beanstalk.response.DescribeConfigurationSettingsResponse (response)
class boto.beanstalk.response.DescribeEnvironmentResourcesResponse (response)
class boto.beanstalk.response.DescribeEnvironmentsResponse (response)
class boto.beanstalk.response.DescribeEventsResponse (response)
class boto.beanstalk.response.EnvironmentDescription (response)
class boto.beanstalk.response.EnvironmentInfoDescription (response)
class boto.beanstalk.response.EnvironmentResourceDescription (response)
class boto.beanstalk.response.EnvironmentResourcesDescription (response)
class boto.beanstalk.response.EventDescription (response)
class boto.beanstalk.response.Instance (response)
class boto.beanstalk.response.LaunchConfiguration (response)
class boto.beanstalk.response.ListAvailableSolutionStacksResponse (response)
class boto.beanstalk.response.Listener (response)
class boto.beanstalk.response.LoadBalancer (response)
class boto.beanstalk.response.LoadBalancerDescription (response)
class boto.beanstalk.response.OptionRestrictionRegex (response)
class boto.beanstalk.response.RebuildEnvironmentResponse (response)
class boto.beanstalk.response.RequestEnvironmentInfoResponse (response)
class boto.beanstalk.response.Response (response)
class boto.beanstalk.response.ResponseMetadata (response)
class boto.beanstalk.response.RestartAppServerResponse (response)
class boto.beanstalk.response.RetrieveEnvironmentInfoResponse (response)
class boto.beanstalk.response.S3Location (response)
class boto.beanstalk.response.SolutionStackDescription (response)
class boto.beanstalk.response.SwapEnvironmentCNAMEsResponse (response)
```

```
class boto.beanstalk.response.SwapEnvironmentCnamesResponse (response)
class boto.beanstalk.response.TerminateEnvironmentResponse (response)
class boto.beanstalk.response.Trigger (response)
class boto.beanstalk.response.UpdateApplicationResponse (response)
class boto.beanstalk.response.UpdateApplicationVersionResponse (response)
class boto.beanstalk.response.UpdateConfigurationTemplateResponse (response)
class boto.beanstalk.response.UpdateEnvironmentResponse (response)
class boto.beanstalk.response.ValidateConfigurationSettingsResponse (response)
class boto.beanstalk.response.ValidationMessage (response)
```

4.109 boto

4.109.1 boto

```
class boto.NullHandler (level=0)
    Initializes the instance - basically setting the formatter to None and the filter list to empty.

    emit (record)
        Do whatever it takes to actually log the specified logging record.

        This version is intended to be implemented by subclasses and so raises a NotImplementedError.

boto.connect_autoscale (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)

    Parameters

        • aws_access_key_id (string) – Your AWS Access Key ID
        • aws_secret_access_key (string) – Your AWS Secret Access Key

    Return type boto.ec2.autoscale.AutoScaleConnection

    Returns A connection to Amazon's Auto Scaling Service

    :type use_block_device_types bool :param use_block_device_types: Specifies whether to return described
    Launch Configs with block device mappings containing

        block device types, or a list of old style block device mappings (deprecated). This defaults to false
        for compatability with the old incorrect style.

boto.connect_awslambda (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)
    Connect to AWS Lambda

    Parameters

        • aws_access_key_id (string) – Your AWS Access Key ID
        • aws_secret_access_key (string) – Your AWS Secret Access Key

    rtype: boto.awslambda.layer1.AWSLambdaConnection :return: A connection to the AWS Lambda
    service

boto.connect_beanstalk (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)

    Parameters

        • aws_access_key_id (string) – Your AWS Access Key ID
```

- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.beanstalk.layer1.Layer1*

Returns A connection to Amazon’s Elastic Beanstalk service

`boto.connect_cloudformation` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.cloudformation.CloudFormationConnection*

Returns A connection to Amazon’s CloudFormation Service

`boto.connect_cloudfront` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.fps.connection.FPSConnection*

Returns A connection to FPS

`boto.connect_cloudhsm` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)
Connect to AWS CloudHSM

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: boto.cloudhsm.layer1.CloudHSMConnection :return: A connection to the AWS CloudHSM service

`boto.connect_cloudsearch` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.cloudsearch.layer2.Layer2*

Returns A connection to Amazon’s CloudSearch service

`boto.connect_cloudsearch2` (*aws_access_key_id=None, aws_secret_access_key=None, sign_request=False, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key
- **sign_request** (*bool*) – whether or not to sign search and upload requests

Return type *boto.cloudsearch2.layer2.Layer2*

Returns A connection to Amazon’s CloudSearch2 service

`boto.connect_cloudsearchdomain` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID
- `aws_secret_access_key` (*string*) – Your AWS Secret Access Key

Return type `boto.cloudsearchdomain.layer1.CloudSearchDomainConnection`

Returns A connection to Amazon’s CloudSearch Domain service

`boto.connect_cloudtrail` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to AWS CloudTrail

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID
- `aws_secret_access_key` (*string*) – Your AWS Secret Access Key

Return type `boto.cloudtrail.layer1.CloudtrailConnection`

Returns A connection to the AWS Cloudtrail service

`boto.connect_cloudwatch` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID
- `aws_secret_access_key` (*string*) – Your AWS Secret Access Key

Return type `boto.ec2.cloudwatch.CloudWatchConnection`

Returns A connection to Amazon’s EC2 Monitoring service

`boto.connect_codedeploy` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to AWS CodeDeploy

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID
- `aws_secret_access_key` (*string*) – Your AWS Secret Access Key

rtype: `boto.cognito.sync.layer1.CodeDeployConnection` **:return:** A connection to the AWS CodeDeploy service

`boto.connect_cognito_identity` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to Amazon Cognito Identity

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID
- `aws_secret_access_key` (*string*) – Your AWS Secret Access Key

rtype: `boto.cognito.identity.layer1.CognitoIdentityConnection` **:return:** A connection to the Amazon Cognito Identity service

`boto.connect_cognito_sync` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to Amazon Cognito Sync

Parameters

- `aws_access_key_id` (*string*) – Your AWS Access Key ID

- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: `boto.cognito.sync.layer1.CognitoSyncConnection` :return: A connection to the Amazon Cognito Sync service

`boto.connect_configservice` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)
Connect to AWS Config

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: `boto.kms.layer1.ConfigServiceConnection` :return: A connection to the AWS Config service

`boto.connect_directconnect` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)
Connect to AWS DirectConnect

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.directconnect.layer1.DirectConnectConnection`

Returns A connection to the AWS DirectConnect service

`boto.connect_dynamodb` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.dynamodb.layer2.Layer2`

Returns A connection to the Layer2 interface for DynamoDB.

`boto.connect_ec2` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ec2.connection.EC2Connection`

Returns A connection to Amazon's EC2

`boto.connect_ec2_endpoint` (*url, aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to an EC2 Api endpoint. Additional arguments are passed through to `connect_ec2`.

Parameters

- **url** (*string*) – A url for the ec2 api endpoint to connect to
- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ec2.connection.EC2Connection`

Returns A connection to Eucalyptus server

`boto.connect_ec2containerservice` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to Amazon EC2 Container Service rtype: `boto.ec2containerservice.layer1.EC2ContainerServiceConnection`:return: A connection to the Amazon EC2 Container Service

`boto.connect_elastictranscoder` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ets.layer1.ElasticTranscoderConnection`

Returns A connection to Amazon's Elastic Transcoder service

`boto.connect_elb` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ec2.elb.ELBConnection`

Returns A connection to Amazon's Load Balancing Service

`boto.connect_emr` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.emr.EmrConnection`

Returns A connection to Elastic mapreduce

`boto.connect_euca` (*host=None, aws_access_key_id=None, aws_secret_access_key=None, port=8773, path='/services/Eucalyptus', is_secure=False, **kwargs*)

Connect to a Eucalyptus service.

Parameters

- **host** (*string*) – the host name or ip address of the Eucalyptus server
- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ec2.connection.EC2Connection`

Returns A connection to Eucalyptus server

`boto.connect_fps` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.fps.connection.FPSConnection`

Returns A connection to FPS

`boto.connect_glacier` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.glacier.layer2.Layer2`

Returns A connection to Amazon’s Glacier Service

`boto.connect_gs` (*gs_access_key_id=None, gs_secret_access_key=None, **kwargs*)

@type `gs_access_key_id`: `string` @param `gs_access_key_id`: Your Google Cloud Storage Access Key ID

@type `gs_secret_access_key`: `string` @param `gs_secret_access_key`: Your Google Cloud Storage Secret Access Key

@rtype: `L{GSConnection<boto.gs.connection.GSConnection>}` @return: A connection to Google’s Storage service

`boto.connect_ia` (*ia_access_key_id=None, ia_secret_access_key=None, is_secure=False, **kwargs*)

Connect to the Internet Archive via their S3-like API.

Parameters

- **ia_access_key_id** (*string*) – Your IA Access Key ID. This will also look in your boto config file for an entry in the Credentials section called “`ia_access_key_id`”
- **ia_secret_access_key** (*string*) – Your IA Secret Access Key. This will also look in your boto config file for an entry in the Credentials section called “`ia_secret_access_key`”

Return type `boto.s3.connection.S3Connection`

Returns A connection to the Internet Archive

`boto.connect_iam` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.iam.IAMConnection`

Returns A connection to Amazon’s IAM

`boto.connect_kinesis` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to Amazon Kinesis

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: `boto.kinesis.layer1.KinesisConnection` :return: A connection to the Amazon Kinesis service

`boto.connect_kms` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Connect to AWS Key Management Service

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: *boto.kms.layer1.KMSConnection* :return: A connection to the AWS Key Management Service

`boto.connect_logs` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)
Connect to Amazon CloudWatch Logs

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: *boto.kinesis.layer1.CloudWatchLogsConnection* :return: A connection to the Amazon CloudWatch Logs service

`boto.connect_machinelearning` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)
Connect to Amazon Machine Learning service rtype: *boto.machinelearning.layer1.MachineLearningConnection* :return: A connection to the Amazon Machine Learning service

`boto.connect_mturk` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.mturk.connection.MTurkConnection*

Returns A connection to MTurk

`boto.connect_opsworks` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

`boto.connect_rds` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.rds.RDSConnection*

Returns A connection to RDS

`boto.connect_rds2` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.rds2.layer1.RDSConnection*

Returns A connection to RDS

`boto.connect_redshift` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.redshift.layer1.RedshiftConnection*

Returns A connection to Amazon's Redshift service

`boto.connect_route53` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.dns.Route53Connection`

Returns A connection to Amazon's Route53 DNS Service

```
boto.connect_route53domains (aws_access_key_id=None,          aws_secret_access_key=None,
                             **kwargs)
```

Connect to Amazon Route 53 Domains

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

rtype: `boto.route53.domains.layer1.Route53DomainsConnection` :return: A connection to the Amazon Route 53 Domains service

```
boto.connect_s3 (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)
```

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.s3.connection.S3Connection`

Returns A connection to Amazon's S3

```
boto.connect_sdb (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)
```

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.sdb.connection.SDBConnection`

Returns A connection to Amazon's SDB

```
boto.connect_ses (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)
```

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.ses.SESConnection`

Returns A connection to Amazon's SES

```
boto.connect_sns (aws_access_key_id=None, aws_secret_access_key=None, **kwargs)
```

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.sns.SNSConnection`

Returns A connection to Amazon's SNS

`boto.connect_sqs` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.sqs.connection.SQSConnection*

Returns A connection to Amazon’s SQS

`boto.connect_sts` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.sts.STSConnection*

Returns A connection to Amazon’s STS

`boto.connect_support` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.support.layer1.SupportConnection*

Returns A connection to Amazon’s Support service

`boto.connect_swf` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.swf.layer1.Layer1*

Returns A connection to the Layer1 interface for SWF.

`boto.connect_vpc` (*aws_access_key_id=None, aws_secret_access_key=None, **kwargs*)

Parameters

- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type *boto.vpc.VPCConnection*

Returns A connection to VPC

`boto.connect_walrus` (*host=None, aws_access_key_id=None, aws_secret_access_key=None, port=8773, path='/services/Walrus', is_secure=False, **kwargs*)

Connect to a Walrus service.

Parameters

- **host** (*string*) – the host name or ip address of the Walrus server
- **aws_access_key_id** (*string*) – Your AWS Access Key ID
- **aws_secret_access_key** (*string*) – Your AWS Secret Access Key

Return type `boto.s3.connection.S3Connection`

Returns A connection to Walrus

`boto.init_logging()`

`boto.set_file_logger(name, filepath, level=20, format_string=None)`

`boto.set_stream_logger(name, level=10, format_string=None)`

`boto.storage_uri(uri_str, default_scheme='file', debug=0, validate=True, bucket_storage_uri_class=<class 'boto.storage_uri.BucketStorageUri'>, suppress_consec_slashes=True, is_latest=False)`

Instantiate a StorageUri from a URI string.

Parameters

- **uri_str** (*string*) – URI naming bucket + optional object.
- **default_scheme** (*string*) – default scheme for scheme-less URIs.
- **debug** (*int*) – debug level to pass in to boto connection (range 0..2).
- **validate** (*bool*) – whether to check for bucket name validity.
- **bucket_storage_uri_class** (*BucketStorageUri interface.*) – Allows mocking for unit tests.
- **suppress_consec_slashes** – If provided, controls whether consecutive slashes will be suppressed in key paths.
- **is_latest** (*bool*) – whether this versioned object represents the current version.

We allow `validate` to be disabled to allow caller to implement bucket-level wildcarding (outside the boto library; see `gsutil`).

Return type `boto.StorageUri` subclass

Returns `StorageUri` subclass for given URI.

`uri_str` must be one of the following formats:

- `gs://bucket/name`
- `gs://bucket/name#ver`
- `s3://bucket/name`
- `gs://bucket`
- `s3://bucket`
- filename (which could be a Unix path like `/a/b/c` or a Windows path like `C:c`)

The last example uses the default scheme ('file', unless overridden).

`boto.storage_uri_for_key(key)`

Returns a `StorageUri` for the given key.

Parameters **key** (`boto.s3.key.Key` or subclass) – URI naming bucket + optional object.

4.109.2 boto.connection

Handles basic connections to AWS

```
class boto.connection.AWSAuthConnection (host, aws_access_key_id=None,
aws_secret_access_key=None, is_secure=True,
port=None, proxy=None, proxy_port=None,
proxy_user=None, proxy_pass=None, debug=0,
https_connection_factory=None, path='/',
provider='aws', security_token=None, sup-
press_consec_slashes=True, validate_certs=True,
profile_name=None)
```

Parameters

- **host** (*str*) – The host to make the connection to
- **aws_access_key_id** (*str*) – Your AWS Access Key ID (provided by Amazon). If none is specified, the value in your `AWS_ACCESS_KEY_ID` environmental variable is used.
- **aws_secret_access_key** (*str*) – Your AWS Secret Access Key (provided by Amazon). If none is specified, the value in your `AWS_SECRET_ACCESS_KEY` environmental variable is used.
- **security_token** (*str*) – The security token associated with temporary credentials issued by STS. Optional unless using temporary credentials. If none is specified, the environment variable `AWS_SECURITY_TOKEN` is used if defined.
- **is_secure** (*boolean*) – Whether the connection is over SSL
- **https_connection_factory** (*list or tuple*) – A pair of an HTTP connection factory and the exceptions to catch. The factory should have a similar interface to `L{http_client.HTTPSConnection}`.
- **proxy** (*str*) – Address/hostname for a proxy server
- **proxy_port** (*int*) – The port to use when connecting over a proxy
- **proxy_user** (*str*) – The username to connect with on the proxy
- **proxy_pass** (*str*) – The password to use when connection over a proxy.
- **port** (*int*) – The port to use to connect
- **suppress_consec_slashes** (*bool*) – If provided, controls whether consecutive slashes will be suppressed in key paths.
- **validate_certs** (*bool*) – Controls whether SSL certificates will be validated or not. Defaults to True.
- **profile_name** (*str*) – Override usual Credentials section in config file to use a named set of keys instead.

access_key

auth_region_name

auth_service_name

aws_access_key_id

aws_secret_access_key

build_base_http_request (*method, path, auth_path, params=None, headers=None, data=""*,
host=None)

close ()

(Optional) Close any open HTTP connections. This is non-destructive, and making a new request will open a connection again.

```

connection
get_http_connection (host, port, is_secure)
get_path (path='/')
get_proxy_auth_header ()
get_proxy_url_with_auth ()
gs_access_key_id
gs_secret_access_key
handle_proxy (proxy, proxy_port, proxy_user, proxy_pass)
make_request (method, path, headers=None, data="", host=None, auth_path=None, sender=None,
                override_num_retries=None, params=None, retry_handler=None)
    Makes a request to the server, with stock multiple-retry logic.
new_http_connection (host, port, is_secure)
prefix_proxy_to_path (path, host=None)
profile_name
proxy_ssl (host=None, port=None)
put_http_connection (host, port, is_secure, connection)
secret_key
server_name (port=None)
set_host_header (request)
set_request_hook (hook)
skip_proxy (host)

class boto.connection.AWSQueryConnection (aws_access_key_id=None,
                                           aws_secret_access_key=None,
                                           is_secure=True, port=None, proxy=None,
                                           proxy_port=None, proxy_user=None,
                                           proxy_pass=None, host=None, debug=0,
                                           https_connection_factory=None, path='/',
                                           security_token=None, validate_certs=True,
                                           profile_name=None, provider='aws')

APIVersion = ''

ResponseError
    alias of boto.exception.BotoServerError

build_complex_list_params (params, items, label, names)
    Serialize a list of structures.

    For example:

```

```

items = [('foo', 'bar', 'baz'), ('foo2', 'bar2', 'baz2')]
label = 'ParamName.member'
names = ('One', 'Two', 'Three')
self.build_complex_list_params(params, items, label, names)

```

would result in the params dict being updated with these params:

```
ParamName.member.1.One = foo
ParamName.member.1.Two = bar
ParamName.member.1.Three = baz

ParamName.member.2.One = foo2
ParamName.member.2.Two = bar2
ParamName.member.2.Three = baz2
```

Parameters

- **params** (*dict*) – The params dict. The complex list params will be added to this dict.
- **items** (*list of tuples*) – The list to serialize.
- **label** (*string*) – The prefix to apply to the parameter.
- **names** (*tuple of strings*) – The names associated with each tuple element.

build_list_params (*params, items, label*)

get_list (*action, params, markers, path='/', parent=None, verb='GET'*)

get_object (*action, params, cls, path='/', parent=None, verb='GET'*)

get_status (*action, params, path='/', parent=None, verb='GET'*)

get_utf8_value (*value*)

make_request (*action, params=None, path='/', verb='GET'*)

Makes a request to the server, with stock multiple-retry logic.

class boto.connection.**ConnectionPool**

A connection pool that expires connections after a fixed period of time. This saves time spent waiting for a connection that AWS has timed out on the other end.

This class is thread-safe.

CLEAN_INTERVAL = 5.0

STALE_DURATION = 60.0

clean ()

Clean up the stale connections in all of the pools, and then get rid of empty pools. Pools clean themselves every time a connection is fetched; this cleaning takes care of pools that aren't being used any more, so nothing is being gotten from them.

get_http_connection (*host, port, is_secure*)

Gets a connection from the pool for the named host. Returns None if there is no connection that can be reused. It's the caller's responsibility to call close() on the connection when it's no longer needed.

put_http_connection (*host, port, is_secure, conn*)

Adds a connection to the pool of connections that can be reused for the named host.

size ()

Returns the number of connections in the pool.

class boto.connection.**HTTPRequest** (*method, protocol, host, port, path, auth_path, params, headers, body*)

Represents an HTTP request.

Parameters

- **method** (*string*) – The HTTP method name, 'GET', 'POST', 'PUT' etc.

- **protocol** (*string*) – The http protocol used, ‘http’ or ‘https’.
- **host** (*string*) – Host to which the request is addressed. eg. abc.com
- **port** (*int*) – port on which the request is being sent. Zero means unset, in which case default port will be chosen.
- **path** (*string*) – URL path that is being accessed.
- **path** – The part of the URL path used when creating the authentication string.
- **params** (*dict*) – HTTP url query parameters, with key as name of the param, and value as value of param.
- **headers** (*dict*) – HTTP headers, with key as name of the header and value as value of header.
- **body** (*string*) – Body of the HTTP request. If not present, will be None or empty string (‘’).

authorize (*connection*, ***kwargs*)

class boto.connection.HTTPResponse (**args*, ***kwargs*)

read (*amt=None*)

Read the response.

This method does not have the same behavior as `http_client.HTTPResponse.read`. Instead, if this method is called with no `amt` arg, then the response body will be cached. Subsequent calls to `read()` with no args **will return the cached response**.

class boto.connection.HostConnectionPool

A pool of connections for one remote (host,port,is_secure).

When connections are added to the pool, they are put into a pending queue. The `_mexe` method returns connections to the pool before the response body has been read, so they connections aren’t ready to send another request yet. They stay in the pending queue until they are ready for another request, at which point they are returned to the pool of ready connections.

The pool of ready connections is an ordered list of (connection,time) pairs, where the time is the time the connection was returned from `_mexe`. After a certain period of time, connections are considered stale, and discarded rather than being reused. This saves having to wait for the connection to time out if AWS has decided to close it on the other end because of inactivity.

Thread Safety:

This class is used only from `ConnectionPool` while it’s mutex is held.

clean ()

Get rid of stale connections.

get ()

Returns the next connection in this pool that is ready to be reused. Returns None if there aren’t any.

put (*conn*)

Adds a connection to the pool, along with the time it was added.

size ()

Returns the number of connections in the pool for this host. Some of the connections may still be in use, and may not be ready to be returned by `get()`.

4.109.3 boto.exception

Exception classes - Subclassing allows you to check for specific errors

exception `boto.exception.AWSConnectionError` (*reason, *args*)
General error connecting to Amazon Web Services.

exception `boto.exception.BotoClientError` (*reason, *args*)
General Boto Client error (error accessing AWS)

exception `boto.exception.BotoServerError` (*status, reason, body=None, *args*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.exception.ConsoleOutput` (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

exception `boto.exception.DynamoDBResponseError` (*status, reason, body=None, *args*)

exception `boto.exception.EC2ResponseError` (*status, reason, body=None*)
Error in response from EC2.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

exception `boto.exception.EmrResponseError` (*status, reason, body=None, *args*)
Error in response from EMR

exception `boto.exception.GSCopyError` (*status, reason, body=None, *args*)
Error copying a key on GS.

exception `boto.exception.GSCreateError` (*status, reason, body=None*)
Error creating a bucket or key on GS.

exception `boto.exception.GSDataError` (*reason, *args*)
Error receiving data from GS.

exception `boto.exception.GSPermissionsError` (*reason, *args*)
Permissions error when accessing a bucket or key on GS.

exception `boto.exception.GSResponseError` (*status, reason, body=None*)
Error in response from GS.

exception `boto.exception.InvalidAclError` (*message*)
Exception raised when ACL XML is invalid.

exception `boto.exception.InvalidCorsError` (*message*)
Exception raised when CORS XML is invalid.

exception `boto.exception.InvalidEncryptionConfigError` (*message*)
Exception raised when GCS encryption configuration XML is invalid.

exception `boto.exception.InvalidInstanceMetadataError` (*msg*)

MSG = "You can set the 'metadata_service_num_attempts' in your boto config file to inc

exception `boto.exception.InvalidLifecycleConfigError` (*message*)
Exception raised when GCS lifecycle configuration XML is invalid.

exception `boto.exception.InvalidUriError` (*message*)
Exception raised when URI is invalid.

exception `boto.exception.JSONResponseError` (*status, reason, body=None, *args*)
This exception expects the fully parsed and decoded JSON response body to be passed as the body parameter.

Variables

- **status** – The HTTP status code.
- **reason** – The HTTP reason message.
- **body** – The Python dict that represents the decoded JSON response body.
- **error_message** – The full description of the AWS error encountered.
- **error_code** – A short string that identifies the AWS error (e.g. ConditionalCheckFailedException)

exception `boto.exception.NoAuthHandlerFound`
Is raised when no auth handlers were found ready to authenticate.

exception `boto.exception.PleaseRetryException` (*message, response=None*)
Indicates a request should be retried.

exception `boto.exception.ResumableDownloadException` (*message, disposition*)
Exception raised for various resumable download problems.
`self.disposition` is of type `ResumableTransferDisposition`.

class `boto.exception.ResumableTransferDisposition`

`ABORT = 'ABORT'`

`ABORT_CUR_PROCESS = 'ABORT_CUR_PROCESS'`

`START_OVER = 'START_OVER'`

`WAIT_BEFORE_RETRY = 'WAIT_BEFORE_RETRY'`

exception `boto.exception.ResumableUploadException` (*message, disposition*)
Exception raised for various resumable upload problems.
`self.disposition` is of type `ResumableTransferDisposition`.

exception `boto.exception.S3CopyError` (*status, reason, body=None, *args*)
Error copying a key on S3.

exception `boto.exception.S3CreateError` (*status, reason, body=None*)
Error creating a bucket or key on S3.

exception `boto.exception.S3DataError` (*reason, *args*)
Error receiving data from S3.

exception `boto.exception.S3PermissionsError` (*reason, *args*)
Permissions error when accessing a bucket or key on S3.

exception `boto.exception.S3ResponseError` (*status, reason, body=None*)
Error in response from S3.

exception `boto.exception.SDBPersistenceError`

exception `boto.exception.SDBResponseError` (*status, reason, body=None, *args*)
Error in responses from SDB.

exception `boto.exception.SQSDecodeError` (*reason, message*)
Error when decoding an SQS message.

exception `boto.exception.SQSError` (*status, reason, body=None*)
General Error on Simple Queue Service.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

exception `boto.exception.SWFResponseError` (*status, reason, body=None, *args*)

exception `boto.exception.StorageCopyError` (*status, reason, body=None, *args*)
Error copying a key on a storage service.

exception `boto.exception.StorageCreateError` (*status, reason, body=None*)
Error creating a bucket or key on a storage service.

endElement (*name, value, connection*)

exception `boto.exception.StorageDataError` (*reason, *args*)
Error receiving data from a storage service.

exception `boto.exception.StoragePermissionsError` (*reason, *args*)
Permissions error when accessing a bucket or key on a storage service.

exception `boto.exception.StorageResponseError` (*status, reason, body=None*)
Error in response from a storage service.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

exception `boto.exception.TooManyRecordsException` (*message*)
Exception raised when a search of Route53 records returns more records than requested.

4.109.4 boto.handler

class `boto.handler.XmlHandler` (*root_node, connection*)

characters (*content*)

endElement (*name*)

startElement (*name, attrs*)

class `boto.handler.XmlHandlerWrapper` (*root_node, connection*)

parseString (*content*)

4.109.5 boto.resultset

class `boto.resultset.BooleanResult` (*marker_elem=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_boolean (*value*, *true_value*='true')

class boto.resultset.**ResultSet** (*marker_elem*=None)

The ResultSet is used to pass results back from the Amazon services to the client. It is light wrapper around Python's `list` class, with some additional methods for parsing XML results from AWS. Because I don't really want any dependencies on external libraries, I'm using the standard SAX parser that comes with Python. The good news is that it's quite fast and efficient but it makes some things rather difficult.

You can pass in, as the `marker_elem` parameter, a list of tuples. Each tuple contains a string as the first element which represents the XML element that the resultset needs to be on the lookout for and a Python class as the second element of the tuple. Each time the specified element is found in the XML, a new instance of the class will be created and popped onto the stack.

Variables `next_token` (*str*) – A hash used to assist in paging through very long result sets. In most cases, passing this value to certain methods will give you another 'page' of results.

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

to_boolean (*value*, *true_value*='true')

4.109.6 boto.utils

Some handy utility functions used by several classes.

class boto.utils.**AuthSMTPHandler** (*mailhost*, *username*, *password*, *fromaddr*, *toaddrs*, *subject*)

This class extends the SMTPHandler in the standard Python logging module to accept a username and password on the constructor and to then use those credentials to authenticate with the SMTP server. To use this, you could add something like this in your boto config file:

```
[handler_hand07] class=boto.utils.AuthSMTPHandler level=WARN formatter=form07 args=('localhost', 'username', 'password', 'from@abc', ['user1@abc', 'user2@xyz'], 'Logger Subject')
```

Initialize the handler.

We have extended the constructor to accept a username/password for SMTP authentication.

emit (*record*)

Emit a record.

Format the record and send it to the specified addressees. It would be really nice if I could add authorization to this class without having to resort to cut and paste inheritance but, no.

class boto.utils.**LRUCache** (*capacity*)

A dictionary-like object that stores only a certain number of items, and discards its least recently used item when full.

```
>>> cache = LRUCache(3)
>>> cache['A'] = 0
>>> cache['B'] = 1
>>> cache['C'] = 2
>>> len(cache)
3
```

```
>>> cache['A']
0
```

Adding new items to the cache does not increase its size. Instead, the least recently used item is dropped:

```
>>> cache['D'] = 3
>>> len(cache)
3
>>> 'B' in cache
False
```

Iterating over the cache returns the keys, starting with the most recently used:

```
>>> for key in cache:
...     print key
D
A
C
```

This code is based on the LRUCache class from Genshi which is based on [Myghty's LRUCache](#) from `myghtyutils.util`, written by Mike Bayer and released under the MIT license (Genshi uses the BSD License).

class `boto.utils.LazyLoadMetadata` (*url, num_retries, timeout=None*)

get (*k, d*) → `D[k]` if `k` in `D`, else `d`. `d` defaults to `None`.

items () → list of `D`'s (key, value) pairs, as 2-tuples

values () → list of `D`'s values

class `boto.utils.Password` (*str=None, hashfunc=None*)

Password object that stores itself as hashed. Hash defaults to SHA512 if available, MD5 otherwise.

Load the string from an initial value, this should be the raw hashed password.

hashfunc ()

Returns a sha512 hash object; optionally initialized with a string

set (*value*)

class `boto.utils.RequestHook`

This can be extended and supplied to the connection object to gain access to request and response object after the request completes. One use for this would be to implement some specific request logging.

handle_request_data (*request, response, error=False*)

class `boto.utils.ShellCommand` (*command, wait=True, fail_fast=False, cwd=None*)

getOutput ()

getStatus ()

output

The STDIN and STDERR output of the command

run (*cwd=None*)

setReadOnly (*value*)

status

The exit code for the command

`boto.utils.canonical_string` (*method, path, headers, expires=None, provider=None*)

Generates the aws canonical string for the given parameters

`boto.utils.compute_hash` (*fp*, *buf_size*=8192, *size*=None, *hash_algorithm*=<built-in function openssl_md5>)

`boto.utils.compute_md5` (*fp*, *buf_size*=8192, *size*=None)

Compute MD5 hash on passed file and return results in a tuple of values.

Parameters

- **fp** (*file*) – File pointer to the file to MD5 hash. The file pointer will be reset to its current location before the method returns.
- **buf_size** (*integer*) – Number of bytes per read request.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (*fp*). This is useful when uploading a file in multiple parts where the file is being split in place into different parts. Less bytes may be available.

Return type tuple

Returns A tuple containing the hex digest version of the MD5 hash as the first element, the base64 encoded version of the plain digest as the second element and the data size as the third element.

`boto.utils.fetch_file` (*uri*, *file*=None, *username*=None, *password*=None)

Fetch a file based on the URI provided. If you do not pass in a file pointer a `tempfile.NamedTemporaryFile`, or None if the file could not be retrieved is returned. The URI can be either an HTTP url, or “s3://bucket_name/key_name”

`boto.utils.find_class` (*module_name*, *class_name*=None)

`boto.utils.find_matching_headers` (*name*, *headers*)

Takes a specific header name and a dict of headers {“name”: “value”}. Returns a list of matching header names, case-insensitive.

`boto.utils.get_aws_metadata` (*headers*, *provider*=None)

`boto.utils.get_instance_identity` (*version*=‘latest’, *url*=‘http://169.254.169.254’, *timeout*=None, *num_retries*=5)

Returns the instance identity as a nested Python dictionary.

`boto.utils.get_instance_metadata` (*version*=‘latest’, *url*=‘http://169.254.169.254’, *data*=‘meta-data’, *timeout*=None, *num_retries*=5)

Returns the instance metadata as a nested Python dictionary. Simple values (e.g. `local_hostname`, `hostname`, etc.) will be stored as string values. Values such as `ancestor-ami-ids` will be stored in the dict as a list of string values. More complex fields such as `public-keys` and will be stored as nested dicts.

If the timeout is specified, the connection to the specified url will time out after the specified number of seconds.

`boto.utils.get_instance_userdata` (*version*=‘latest’, *sep*=None, *url*=‘http://169.254.169.254’, *timeout*=None, *num_retries*=5)

`boto.utils.get_ts` (*ts*=None)

`boto.utils.get_utf8_value` (*value*)

`boto.utils.guess_mime_type` (*content*, *deftype*)

Description: Guess the mime type of a block of text :param content: content we’re finding the type of :type str:

Parameters **deftype** – Default mime type

Return type <type>:

Returns <description>

`boto.utils.host_is_ipv6` (*hostname*)

Detect (naively) if the hostname is an IPV6 host. Return a boolean.

`boto.utils.merge_headers_by_name` (*name, headers*)

Takes a specific header name and a dict of headers {"name": "value"}. Returns a string of all header values, comma-separated, that match the input header name, case-insensitive.

`boto.utils.merge_meta` (*headers, metadata, provider=None*)

`boto.utils.mklist` (*value*)

`boto.utils.notify` (*subject, body=None, html_body=None, to_string=None, attachments=None, append_instance_id=True*)

`boto.utils.parse_host` (*hostname*)

Given a hostname that may have a port name, ensure that the port is trimmed returning only the host, including hostnames that are IPV6 and may include brackets.

`boto.utils.parse_ts` (*ts*)

`boto.utils.pythonize_name` (*name*)

Convert camel case to a "pythonic" name.

Examples:

```
pythonize_name('CamelCase') -> 'camel_case'
pythonize_name('already_pythonized') -> 'already_pythonized'
pythonize_name('HTTPRequest') -> 'http_request'
pythonize_name('HTTPStatus200Ok') -> 'http_status_200_ok'
pythonize_name('UPPER') -> 'upper'
pythonize_name('') -> ''
```

`boto.utils.retry_url` (*url, retry_on_404=True, num_retries=10, timeout=None*)

Retry a url. This is specifically used for accessing the metadata service on an instance. Since this address should never be proxied (for security reasons), we create a ProxyHandler with a NULL dictionary to override any proxy settings in the environment.

`boto.utils.setlocale` (**args, **kws*)

A context manager to set the locale in a threadsafe manner.

`boto.utils.unquote_v` (*nv*)

`boto.utils.update_dme` (*username, password, dme_id, ip_address*)

Update your Dynamic DNS record with DNSMadeEasy.com

`boto.utils.write_mime_multipart` (*content, compress=False, deftype='text/plain', delimiter=':'*)

Description: :param content: A list of tuples of name-content pairs. This is used instead of a dict to ensure that scripts run in order :type list of tuples:

Parameters

- **compress** – Use gzip to compress the scripts, defaults to no compression
- **deftype** – The type that should be assumed if nothing else can be figured out
- **delimiter** – mime delimiter

Returns Final mime multipart

Return type str:

4.110 cloudformation

4.110.1 boto.cloudformation

`boto.cloudformation.connect_to_region(region_name, **kw_params)`

Given a valid region name, return a `boto.cloudformation.CloudFormationConnection`.

Parameters `region_name` (*str*) – The name of the region to connect to.

Return type `boto.cloudformation.CloudFormationConnection` or `None`

Returns A connection to the given region, or `None` if an invalid region name is given

`boto.cloudformation.regions()`

Get all available regions for the CloudFormation service.

Return type `list`

Returns A list of `boto.RegionInfo` instances

4.110.2 boto.cloudformation.connection

class `boto.cloudformation.connection.CloudFormationConnection` (*aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, proxy_user=None, proxy_pass=None, debug=0, https_connection_factory=None, region=None, path='/', converter=None, security_token=None, validate_certs=True, profile_name=None*)

AWS CloudFormation enables you to create and manage AWS infrastructure deployments predictably and repeatedly. AWS CloudFormation helps you leverage AWS products such as Amazon EC2, EBS, Amazon SNS, ELB, and Auto Scaling to build highly-reliable, highly scalable, cost effective applications without worrying about creating and configuring the underlying AWS infrastructure.

With AWS CloudFormation, you declare all of your resources and dependencies in a template file. The template defines a collection of resources as a single unit called a stack. AWS CloudFormation creates and deletes all member resources of the stack together and manages all dependencies between the resources for you.

For more information about this product, go to the [‘CloudFormation Product Page’](#).

Amazon CloudFormation makes use of other AWS products. If you need additional technical information about a specific AWS product, you can find the product’s technical documentation at [‘http://aws.amazon.com/documentation/’](http://aws.amazon.com/documentation/).

`APIVersion = '2010-05-15'`

```
DefaultRegionEndpoint = 'cloudformation.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
cancel_update_stack(stack_name_or_id=None)
```

Cancels an update on the specified stack. If the call completes successfully, the stack will roll back the update and revert to the previous stack configuration. Only stacks that are in the UPDATE_IN_PROGRESS state can be canceled.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

```
create_stack(stack_name, template_body=None, template_url=None, parameters=None,
             notification_arns=None, disable_rollback=None, timeout_in_minutes=None,
             capabilities=None, tags=None, on_failure=None, stack_policy_body=None,
             stack_policy_url=None)
```

Creates a stack as specified in the template. After the call completes successfully, the stack creation starts. You can check the status of the stack via the DescribeStacks API. Currently, the limit for stacks is 20 stacks per account per region.

Parameters `stack_name` (*string*) –

The name associated with the stack. The name must be unique within your AWS account.

Must contain only alphanumeric characters (case sensitive) and start with an alpha character. Maximum length of the name is 255 characters.

Parameters `template_body` (*string*) – Structure containing the template body. (For more information, go to **‘Template Anatomy’** in the AWS CloudFormation User Guide.)

Conditional: You must pass *TemplateBody* or *TemplateURL*. If both are passed, only *TemplateBody* is used.

Parameters `template_url` (*string*) – Location of file containing the template body. The URL must point to a template (max size: 307,200 bytes) located in an S3 bucket in the same region as the stack. For more information, go to the **‘Template Anatomy’** in the AWS CloudFormation User Guide.

Conditional: You must pass *TemplateURL* or *TemplateBody*. If both are passed, only *TemplateBody* is used.

Parameters

- **parameters** (*list*) – A list of key/value tuples that specify input parameters for the stack.
- **disable_rollback** (*boolean*) – Set to *True* to disable rollback of the stack if stack creation failed. You can specify either *DisableRollback* or *OnFailure*, but not both.

Default: *False*

Parameters

- **timeout_in_minutes** (*integer*) – The amount of time that can pass before the stack status becomes CREATE_FAILED; if *DisableRollback* is not set or is set to *False*, the stack will be rolled back.
- **notification_arns** (*list*) – The Simple Notification Service (SNS) topic ARNs to publish stack related events. You can find your SNS topic ARNs using the **‘SNS console’** or your Command Line Interface (CLI).

- **capabilities** (*list*) – The list of capabilities that you want to allow in the stack. If your template contains certain resources, you must specify the `CAPABILITY_IAM` value for this parameter; otherwise, this action returns an `InsufficientCapabilities` error. The following resources require you to specify the capabilities parameter: `'AWS::CloudFormation::Stack'`, `'AWS::IAM::AccessKey'`, `'AWS::IAM::Group'`, `'AWS::IAM::InstanceProfile'`, `'AWS::IAM::Policy'`, `'AWS::IAM::Role'`, `'AWS::IAM::User'`, and `'AWS::IAM::UserToGroupAddition'`.
- **on_failure** (*string*) – Determines what action will be taken if stack creation fails. This must be one of: `DO_NOTHING`, `ROLLBACK`, or `DELETE`. You can specify either *OnFailure* or *DisableRollback*, but not both.

Default: `ROLLBACK`

Parameters `stack_policy_body` (*string*) – Structure containing the stack policy body. (For more information, go to ‘Prevent Updates to Stack Resources’ in the AWS CloudFormation User Guide.)

If you pass `StackPolicyBody` and `StackPolicyURL`, only `StackPolicyBody` is used.

Parameters

- **stack_policy_url** (*string*) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. If you pass `StackPolicyBody` and `StackPolicyURL`, only `StackPolicyBody` is used.
- **tags** (*dict*) – A set of user-defined *Tags* to associate with this stack, represented by key/value pairs. Tags defined for the stack are propagated to EC2 resources that are created as part of the stack. A maximum number of 10 tags can be specified.

`delete_stack` (*stack_name_or_id*)

Deletes a specified stack. Once the call completes successfully, stack deletion starts. Deleted stacks do not show up in the `DescribeStacks` API if the deletion has been completed successfully.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

`describe_stack_events` (*stack_name_or_id=None, next_token=None*)

Returns all stack related events for a specified stack. For more information about a stack’s event history, go to ‘**Stacks**’ in the AWS CloudFormation User Guide. Events are returned, even if the stack never existed or has been successfully deleted.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

Default: There is no default value.

Parameters `next_token` (*string*) – String that identifies the start of the next list of events, if there is one.

Default: There is no default value.

`describe_stack_resource` (*stack_name_or_id, logical_resource_id*)

Returns a description of the specified resource in the specified stack.

For deleted stacks, `DescribeStackResource` returns resource information for up to 90 days after the stack has been deleted.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

Default: There is no default value.

Parameters `logical_resource_id` (*string*) – The logical name of the resource as specified in the template.

Default: There is no default value.

describe_stack_resources (*stack_name_or_id=None, logical_resource_id=None, physical_resource_id=None*)

Returns AWS resource descriptions for running and deleted stacks. If *StackName* is specified, all the associated resources that are part of the stack are returned. If *PhysicalResourceId* is specified, the associated resources of the stack that the resource belongs to are returned. Only the first 100 resources will be returned. If your stack has more resources than this, you should use *ListStackResources* instead. For deleted stacks, *DescribeStackResources* returns resource information for up to 90 days after the stack has been deleted.

You must specify either *StackName* or *PhysicalResourceId*, but not both. In addition, you can specify *LogicalResourceId* to filter the returned result. For more information about resources, the *LogicalResourceId* and *PhysicalResourceId*, go to the ‘[AWS CloudFormation User Guide](#)’. A *ValidationError* is returned if you specify both *StackName* and *PhysicalResourceId* in the same request.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

Required: Conditional. If you do not specify *StackName*, you must specify *PhysicalResourceId*.

Default: There is no default value.

Parameters `logical_resource_id` (*string*) – The logical name of the resource as specified in the template.

Default: There is no default value.

Parameters `physical_resource_id` (*string*) – The name or unique identifier that corresponds to a physical instance ID of a resource supported by AWS CloudFormation.

For example, for an Amazon Elastic Compute Cloud (EC2) instance, *PhysicalResourceId* corresponds to the *InstanceId*. You can pass the EC2 *InstanceId* to *DescribeStackResources* to find which stack the instance belongs to and what other resources are part of the stack.

Required: Conditional. If you do not specify *PhysicalResourceId*, you must specify *StackName*.

Default: There is no default value.

describe_stacks (*stack_name_or_id=None, next_token=None*)

Returns the description for the specified stack; if no stack name was specified, then it returns the description for all the stacks created.

Parameters `stack_name_or_id` (*string*) – The name or the unique identifier associated with the stack.

Default: There is no default value.

Parameters `next_token` (*string*) – String that identifies the start of the next list of stacks, if there is one.

encode_bool (*v*)

estimate_template_cost (*template_body=None, template_url=None, parameters=None*)

Returns the estimated monthly cost of a template. The return value is an AWS Simple Monthly Calculator URL with a query string that describes the resources required to run the template.

Parameters **template_body** (*string*) – Structure containing the template body. (For more information, go to ‘**Template Anatomy**’_ in the AWS CloudFormation User Guide.)

Conditional: You must pass *TemplateBody* or *TemplateURL*. If both are passed, only *TemplateBody* is used.

Parameters **template_url** (*string*) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack. For more information, go to ‘**Template Anatomy**’_ in the AWS CloudFormation User Guide.

Conditional: You must pass *TemplateURL* or *TemplateBody*. If both are passed, only *TemplateBody* is used.

Parameters **parameters** (*list*) – A list of key/value tuples that specify input parameters for the template.

Return type string

Returns URL to pre-filled cost calculator

get_stack_policy (*stack_name_or_id*)

Returns the stack policy for a specified stack. If a stack doesn’t have a policy, a null value is returned.

Parameters **stack_name_or_id** (*string*) – The name or stack ID that is associated with the stack whose policy you want to get.

Return type string

Returns The policy JSON document

get_template (*stack_name_or_id*)

Returns the template body for a specified stack. You can get the template for running or deleted stacks.

For deleted stacks, GetTemplate returns the template for up to 90 days after the stack has been deleted. If the template does not exist, a *ValidationError* is returned.

Parameters **stack_name_or_id** (*string*) – The name or the unique identifier associated with the stack, which are not always interchangeable:

- **Running stacks:** You can specify either the stack’s name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

list_stack_resources (*stack_name_or_id, next_token=None*)

Returns descriptions of all resources of the specified stack.

For deleted stacks, ListStackResources returns resource information for up to 90 days after the stack has been deleted.

Parameters **stack_name_or_id** (*string*) – The name or the unique identifier associated with the stack, which are not always interchangeable:

- **Running stacks:** You can specify either the stack’s name or its unique stack ID.
- Deleted stacks: You must specify the unique stack ID.

Default: There is no default value.

Parameters `next_token` (*string*) – String that identifies the start of the next list of stack resource summaries, if there is one.

Default: There is no default value.

list_stacks (*stack_status_filters=None, next_token=None*)

Returns the summary information for stacks whose status matches the specified StackStatusFilter. Summary information for stacks that have been deleted is kept for 90 days after the stack is deleted. If no StackStatusFilter is specified, summary information for all stacks is returned (including existing stacks and stacks that have been deleted).

Parameters `next_token` (*string*) – String that identifies the start of the next list of stacks, if there is one.

Default: There is no default value.

Parameters `stack_status_filter` (*list*) – Stack status to use as a filter. Specify one or more stack status codes to list only stacks with the specified status codes. For a complete list of stack status codes, see the *StackStatus* parameter of the Stack data type.

set_stack_policy (*stack_name_or_id, stack_policy_body=None, stack_policy_url=None*)

Sets a stack policy for a specified stack.

Parameters

- **stack_name_or_id** (*string*) – The name or stack ID that you want to associate a policy with.
- **stack_policy_body** (*string*) – Structure containing the stack policy body. (For more information, go to ‘Prevent Updates to Stack Resources’ in the AWS CloudFormation User Guide.)

You must pass *StackPolicyBody* or *StackPolicyURL*. If both are passed, only *StackPolicyBody* is used.

Parameters `stack_policy_url` (*string*) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You must pass *StackPolicyBody* or *StackPolicyURL*. If both are passed, only *StackPolicyBody* is used.

update_stack (*stack_name, template_body=None, template_url=None, parameters=None, notification_arns=None, disable_rollback=False, timeout_in_minutes=None, capabilities=None, tags=None, use_previous_template=None, stack_policy_during_update_body=None, stack_policy_during_update_url=None, stack_policy_body=None, stack_policy_url=None*)

Updates a stack as specified in the template. After the call completes successfully, the stack update starts. You can check the status of the stack via the DescribeStacks action.

****Note: ****You cannot update ‘**AWS::S3::Bucket**’ resources, for example, to add or modify tags.

To get a copy of the template for an existing stack, you can use the GetTemplate action.

Tags that were associated with this stack during creation time will still be associated with the stack after an *UpdateStack* operation.

For more information about creating an update template, updating a stack, and monitoring the progress of the update, see ‘**Updating a Stack**’.

Parameters `stack_name` (*string*) –

The name or stack ID of the stack to update.

Must contain only alphanumeric characters (case sensitive) and start with an alpha character. Maximum length of the name is 255 characters.

Parameters `template_body` (*string*) – Structure containing the template body. (For more information, go to **‘Template Anatomy’** in the AWS CloudFormation User Guide.)

Conditional: You must pass either *UsePreviousTemplate* or one of *TemplateBody* or *TemplateUrl*. If both *TemplateBody* and *TemplateUrl* are passed, only *TemplateBody* is used.

Parameters `template_url` (*string*) – Location of file containing the template body. The URL must point to a template (max size: 307,200 bytes) located in an S3 bucket in the same region as the stack. For more information, go to the **‘Template Anatomy’** in the AWS CloudFormation User Guide.

Conditional: You must pass either *UsePreviousTemplate* or one of *TemplateBody* or *TemplateUrl*. If both *TemplateBody* and *TemplateUrl* are passed, only *TemplateBody* is used. *TemplateBody*.

Parameters `use_previous_template` (*boolean*) – Set to *True* to use the previous template instead of uploading a new one via *TemplateBody* or *TemplateURL*.

Conditional: You must pass either *UsePreviousTemplate* or one of *TemplateBody* or *TemplateUrl*.

Parameters

- **parameters** (*list*) – A list of key/value tuples that specify input parameters for the stack. A 3-tuple (key, value, bool) may be used to specify the *UsePreviousValue* option.
- **notification_arns** (*list*) – The Simple Notification Service (SNS) topic ARNs to publish stack related events. You can find your SNS topic ARNs using the **‘SNS console’** or your Command Line Interface (CLI).
- **disable_rollback** (*bool*) – Indicates whether or not to rollback on failure.
- **timeout_in_minutes** (*integer*) – The amount of time that can pass before the stack status becomes `CREATE_FAILED`; if *DisableRollback* is not set or is set to *False*, the stack will be rolled back.
- **capabilities** (*list*) – The list of capabilities you want to allow in the stack. Currently, the only valid capability is `CAPABILITY_IAM`.
- **tags** (*dict*) – A set of user-defined *Tags* to associate with this stack, represented by key/value pairs. Tags defined for the stack are propagated to EC2 resources that are created as part of the stack. A maximum number of 10 tags can be specified.
- **template_url** (*string*) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack. For more information, go to **‘Template Anatomy’** in the AWS CloudFormation User Guide.

Conditional: You must pass *TemplateURL* or *TemplateBody*. If both are passed, only *TemplateBody* is used.

Parameters `stack_policy_during_update_body` (*string*) – Structure containing the temporary overriding stack policy body. If you pass *StackPolicyDuringUpdateBody* and *StackPolicyDuringUpdateURL*, only *StackPolicyDuringUpdateBody* is used.

If you want to update protected resources, specify a temporary overriding stack policy during this update. If you do not specify a stack policy, the current policy that associated with the stack will be used.

Parameters `stack_policy_during_update_url` (*string*) – Location of a file containing the temporary overriding stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. If you pass *StackPolicyDuringUpdateBody* and *StackPolicyDuringUpdateURL*, only *StackPolicyDuringUpdateBody* is used.

If you want to update protected resources, specify a temporary overriding stack policy during this update. If you do not specify a stack policy, the current policy that is associated with the stack will be used.

Return type string

Returns The unique Stack ID.

`valid_states = ('CREATE_IN_PROGRESS', 'CREATE_FAILED', 'CREATE_COMPLETE', 'ROLLBACK_IN`

`validate_template` (*template_body=None, template_url=None*)

Validates a specified template.

Parameters `template_body` (*string*) – String containing the template body. (For more information, go to ‘[Template Anatomy](#)’ in the AWS CloudFormation User Guide.)

Conditional: You must pass *TemplateURL* or *TemplateBody*. If both are passed, only *TemplateBody* is used.

Parameters `template_url` (*string*) – Location of file containing the template body. The URL must point to a template (max size: 307,200 bytes) located in an S3 bucket in the same region as the stack. For more information, go to ‘[Template Anatomy](#)’ in the AWS CloudFormation User Guide.

Conditional: You must pass *TemplateURL* or *TemplateBody*. If both are passed, only *TemplateBody* is used.

4.110.3 boto.cloudformation.stack

`class` `boto.cloudformation.stack.Capability` (*connection=None*)

`endElement` (*name, value, connection*)

`startElement` (*name, attrs, connection*)

`class` `boto.cloudformation.stack.NotificationARN` (*connection=None*)

`endElement` (*name, value, connection*)

`startElement` (*name, attrs, connection*)

`class` `boto.cloudformation.stack.Output` (*connection=None*)

`endElement` (*name, value, connection*)

`startElement` (*name, attrs, connection*)

`class` `boto.cloudformation.stack.Parameter` (*connection=None*)

`endElement` (*name, value, connection*)

```

    startElement (name, attrs, connection)
class boto.cloudformation.stack.Stack (connection=None)

    delete ()
    describe_events (next_token=None)
    describe_resource (logical_resource_id)
    describe_resources (logical_resource_id=None, physical_resource_id=None)
    endElement (name, value, connection)
    get_policy ()
        Returns the stack policy for this stack. If it has no policy then, a null value is returned.
    get_template ()
    list_resources (next_token=None)
    set_policy (stack_policy_body=None, stack_policy_url=None)
        Sets a stack policy for this stack.

        Parameters stack_policy_body (string) – Structure containing the stack policy body.
            (For more information, go to ‘ Prevent Updates to Stack Resources‘_ in the AWS CloudFormation User Guide.)

        You must pass StackPolicyBody or StackPolicyURL. If both are passed, only StackPolicyBody is used.

        Parameters stack_policy_url (string) – Location of a file containing the stack policy.
            The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. You must pass StackPolicyBody or StackPolicyURL. If both are passed, only StackPolicyBody is used.

    stack_name_reason
    startElement (name, attrs, connection)
    update ()
class boto.cloudformation.stack.StackEvent (connection=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)
    valid_states = ('CREATE_IN_PROGRESS', 'CREATE_FAILED', 'CREATE_COMPLETE', 'DELETE_IN_P
class boto.cloudformation.stack.StackResource (connection=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)
class boto.cloudformation.stack.StackResourceSummary (connection=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

```

```
class boto.cloudformation.stack.StackSummary (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.cloudformation.stack.Tag (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.110.4 boto.cloudformation.template

```
class boto.cloudformation.template.Template (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.cloudformation.template.TemplateParameter (parent)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.111 CloudFront

4.111.1 boto.cloudfront

```
class boto.cloudfront.CloudFrontConnection (aws_access_key_id=None,  
                                             aws_secret_access_key=None, port=None,  
                                             proxy=None, proxy_port=None,  
                                             host='cloudfront.amazonaws.com', de-  
                                             bug=0, security_token=None, vali-  
                                             date_certs=True, profile_name=None,  
                                             https_connection_factory=None)
```

```
    DefaultHost = 'cloudfront.amazonaws.com'
```

```
    Version = '2010-11-01'
```

```
    create_distribution (origin, enabled, caller_reference="", cnames=None, comment="",  
                        trusted_signers=None)
```

```
    create_invalidation_request (distribution_id, paths, caller_reference=None)
```

```
        Creates a new invalidation request :see: http://goo.gl/8vECq
```

```
    create_origin_access_identity (caller_reference="", comment="")
```

```
    create_streaming_distribution (origin, enabled, caller_reference="", cnames=None, com-  
                                   ment="", trusted_signers=None)
```

```
    delete_distribution (distribution_id, etag)
```

```
    delete_origin_access_identity (access_id, etag)
```


`delete_streaming_distribution` (*distribution_id*, *etag*)

`get_all_distributions` ()

`get_all_origin_access_identity` ()

`get_all_streaming_distributions` ()

`get_distribution_config` (*distribution_id*)

`get_distribution_info` (*distribution_id*)

`get_etag` (*response*)

`get_invalidation_requests` (*distribution_id*, *marker=None*, *max_items=None*)

Get all invalidation requests for a given CloudFront distribution. This returns an instance of an InvalidationListResultSet that automatically handles all of the result paging, etc. from CF - you just need to keep iterating until there are no more results.

Parameters

- **distribution_id** (*string*) – The id of the CloudFront distribution
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results and only in a follow-up request to indicate the maximum number of invalidation requests you want in the response. You will need to pass the next_marker property from the previous InvalidationListResultSet response in the follow-up request in order to get the next ‘page’ of results.

Return type `boto.cloudfront.invalidation.InvalidationListResultSet`

Returns An InvalidationListResultSet iterator that lists invalidation requests for a given CloudFront distribution. Automatically handles paging the results.

`get_origin_access_identity_config` (*access_id*)

`get_origin_access_identity_info` (*access_id*)

`get_streaming_distribution_config` (*distribution_id*)

`get_streaming_distribution_info` (*distribution_id*)

`invalidation_request_status` (*distribution_id*, *request_id*, *caller_reference=None*)

`set_distribution_config` (*distribution_id*, *etag*, *config*)

`set_origin_access_identity_config` (*access_id*, *etag*, *config*)

`set_streaming_distribution_config` (*distribution_id*, *etag*, *config*)

4.111.2 boto.cloudfront.distribution

```
class boto.cloudfront.distribution.Distribution (connection=None,          config=None,
                                                domain_name="",          id="",
                                                last_modified_time=None, status="")
```

`add_object` (*name*, *content*, *headers=None*, *replace=True*)

Adds a new content object to the Distribution. The content for the object will be copied to a new Key in the S3 Bucket and the permissions will be set appropriately for the type of Distribution.

Parameters

- **name** (*str or unicode*) – The name or key of the new object.
- **content** (*file-like object*) – A file-like object that contains the content for the new object.
- **headers** (*dict*) – A dictionary containing additional headers you would like associated with the new object in S3.

Return type *boto.cloudfront.object.Object*

Returns The newly created object.

create_signed_url (*url, keypair_id, expire_time=None, valid_after_time=None, ip_address=None, policy_url=None, private_key_file=None, private_key_string=None*)
Creates a signed CloudFront URL that is only valid within the specified parameters.

Parameters

- **url** (*str*) – The URL of the protected object.
- **keypair_id** (*str*) – The keypair ID of the Amazon KeyPair used to sign the URL. This ID MUST correspond to the private key specified with `private_key_file` or `private_key_string`.
- **expire_time** (*int*) – The expiry time of the URL. If provided, the URL will expire after the time has passed. If not provided the URL will never expire. Format is a unix epoch. Use `int(time.time() + duration_in_sec)`.
- **valid_after_time** (*int*) – If provided, the URL will not be valid until after `valid_after_time`. Format is a unix epoch. Use `int(time.time() + secs_until_valid)`.
- **ip_address** (*str*) – If provided, only allows access from the specified IP address. Use `'192.168.0.10'` for a single IP or use `'192.168.0.0/24'` CIDR notation for a subnet.
- **policy_url** (*str*) – If provided, allows the signature to contain wildcard globs in the URL. For example, you could provide: `'http://example.com/media/*'` and the policy and signature would allow access to all contents of the media subdirectory. If not specified, only allow access to the exact url provided in `'url'`.
- **private_key_file** (*str or file object.*) – If provided, contains the filename of the private key file used for signing or an open file object containing the private key contents. Only one of `private_key_file` or `private_key_string` can be provided.
- **private_key_string** (*str*) – If provided, contains the private key string used for signing. Only one of `private_key_file` or `private_key_string` can be provided.

Return type *str*

Returns The signed URL.

delete ()

Delete this CloudFront Distribution. The content associated with the Distribution is not deleted from the underlying Origin bucket in S3.

disable ()

Deactivate the Distribution. A convenience wrapper around the update method.

enable ()

Activate the Distribution. A convenience wrapper around the update method.

endElement (*name, value, connection*)

get_objects ()

Return a list of all content objects in this distribution.

Return type list of `boto.cloudfront.object.Object`

Returns The content objects

set_permissions (*object, replace=False*)

Sets the S3 ACL grants for the given object to the appropriate value based on the type of Distribution. If the Distribution is serving private content the ACL will be set to include the Origin Access Identity associated with the Distribution. If the Distribution is serving public content the content will be set up with “public-read”.

Parameters

- **enabled** – The Object whose ACL is being set
- **replace** (*bool*) – If False, the Origin Access Identity will be appended to the existing ACL for the object. If True, the ACL for the object will be completely replaced with one that grants READ permission to the Origin Access Identity.

set_permissions_all (*replace=False*)

Sets the S3 ACL grants for all objects in the Distribution to the appropriate value based on the type of Distribution.

Parameters **replace** (*bool*) – If False, the Origin Access Identity will be appended to the existing ACL for the object. If True, the ACL for the object will be completely replaced with one that grants READ permission to the Origin Access Identity.

startElement (*name, attrs, connection*)

update (*enabled=None, cnames=None, comment=None*)

Update the configuration of the Distribution. The only values of the DistributionConfig that can be directly updated are:

- CNAMEs
- Comment
- Whether the Distribution is enabled or not

Any changes to the `trusted_signers` or `origin` properties of this distribution’s current config object will also be included in the update. Therefore, to set the origin access identity for this distribution, set `Distribution.config.origin.origin_access_identity` before calling this update method.

Parameters

- **enabled** (*bool*) – Whether the Distribution is active or not.
- **cnames** (*list of str*) – The DNS CNAME’s associated with this Distribution. Maximum of 10 values.
- **comment** (*str or unicode*) – The comment associated with the Distribution.

```
class boto.cloudfront.distribution.DistributionConfig(connection=None, origin=None, enabled=False, caller_reference="", cnames=None, comment="", trusted_signers=None, default_root_object=None, logging=None)
```

Parameters

- **origin** (*boto.cloudfront.origin.S3Origin* or *boto.cloudfront.origin.CustomOrigin*) – Origin information to associate with the distribution. If

your distribution will use an Amazon S3 origin, then this should be an S3Origin object. If your distribution will use a custom origin (non Amazon S3), then this should be a CustomOrigin object.

- **enabled** (*array of str*) – Whether the distribution is enabled to accept end user requests for content.
- **caller_reference** – A unique number that ensures the request can't be replayed. If no caller_reference is provided, boto will generate a type 4 UUID for use as the caller reference.
- **cnames** – A CNAME alias you want to associate with this distribution. You can have up to 10 CNAME aliases per distribution.
- **comment** (*str*) – Any comments you want to include about the distribution.
- **trusted_signers** (*:class`boto.cloudfront.signers.TrustedSigners`*) – Specifies any AWS accounts you want to permit to create signed URLs for private content. If you want the distribution to use signed URLs, this should contain a TrustedSigners object; if you want the distribution to use basic URLs, leave this None.
- **default_root_object** – Designates a default root object. Only include a Default-RootObject value if you are going to assign a default root object for the distribution.
- **logging** (*:class`boto.cloudfront.logging.LoggingInfo`*) – Controls whether access logs are written for the distribution. If you want to turn on access logs, this should contain a LoggingInfo object; otherwise it should contain None.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

```
class boto.cloudfront.distribution.DistributionSummary (connection=None,
                                                         domain_name="", id="",
                                                         last_modified_time=None,
                                                         status="", origin=None,
                                                         cname="", comment="",
                                                         enabled=False)
```

endElement (*name, value, connection*)

get_distribution ()

startElement (*name, attrs, connection*)

```
class boto.cloudfront.distribution.StreamingDistribution (connection=None,
                                                          config=None, domain_name="", id="",
                                                          last_modified_time=None,
                                                          status="")
```

delete ()

Delete this CloudFront Distribution. The content associated with the Distribution is not deleted from the underlying Origin bucket in S3.

startElement (*name, attrs, connection*)

update (*enabled=None, cnames=None, comment=None*)

Update the configuration of the StreamingDistribution. The only values of the StreamingDistributionConfig that can be directly updated are:

- CNAMEs
- Comment
- Whether the Distribution is enabled or not

Any changes to the `trusted_signers` or `origin` properties of this distribution's current config object will also be included in the update. Therefore, to set the origin access identity for this distribution, set `StreamingDistribution.config.origin.origin_access_identity` before calling this update method.

Parameters

- **enabled** (*bool*) – Whether the StreamingDistribution is active or not.
- **cnames** (*list of str*) – The DNS CNAME's associated with this Distribution. Maximum of 10 values.
- **comment** (*str or unicode*) – The comment associated with the Distribution.

```
class boto.cloudfront.distribution.StreamingDistributionConfig (connection=None,
                                                              origin="", enabled=False,
                                                              caller_reference="",
                                                              cnames=None,
                                                              comment="",
                                                              trusted_signers=None,
                                                              logging=None)

    to_xml ()

class boto.cloudfront.distribution.StreamingDistributionSummary (connection=None,
                                                                do-
                                                                main_name="",
                                                                id="",
                                                                last_modified_time=None,
                                                                status="", ori-
                                                                gin=None,
                                                                cname="",
                                                                com-
                                                                ment="", en-
                                                                abled=False)

    get_distribution ()
```

4.111.3 boto.cloudfront.origin

```
class boto.cloudfront.origin.CustomOrigin (dns_name=None, http_port=80,
                                           https_port=443, ori-
                                           gin_protocol_policy=None)
```

Origin information to associate with the distribution. If your distribution will use a non-Amazon S3 origin, then you use the CustomOrigin element.

Parameters

- **dns_name** (*str*) – The DNS name of your Amazon S3 bucket to associate with the distribution. For example: `mybucket.s3.amazonaws.com`.
- **http_port** (*int*) – The HTTP port the custom origin listens on.

- **https_port** – The HTTPS port the custom origin listens on.
- **origin_protocol_policy** (*str*) – The origin protocol policy to apply to your origin. If you specify `http-only`, CloudFront will use HTTP only to access the origin. If you specify `match-viewer`, CloudFront will fetch from your origin using HTTP or HTTPS, based on the protocol of the viewer request.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class `boto.cloudfront.origin.S3Origin` (*dns_name=None, origin_access_identity=None*)
Origin information to associate with the distribution. If your distribution will use an Amazon S3 origin, then you use the S3Origin element.

Parameters

- **dns_name** (*str*) – The DNS name of your Amazon S3 bucket to associate with the distribution. For example: `mybucket.s3.amazonaws.com`.
- **origin_access_identity** (*str*) – The CloudFront origin access identity to associate with the distribution. If you want the distribution to serve private content, include this element; if you want the distribution to serve public content, remove this element.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

`boto.cloudfront.origin.get_oai_value` (*origin_access_identity*)

4.111.4 boto.cloudfront.identity

class `boto.cloudfront.identity.OriginAccessIdentity` (*connection=None, config=None, id="", s3_user_id="", comment=""*)

delete ()

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

update (*comment=None*)

uri ()

class `boto.cloudfront.identity.OriginAccessIdentityConfig` (*connection=None, caller_reference="", comment=""*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class `boto.cloudfront.identity.OriginAccessIdentitySummary` (*connection=None, id="", s3_user_id="", comment=""*)

```

endElement (name, value, connection)
get_origin_access_identity ()
startElement (name, attrs, connection)

```

4.111.5 boto.cloudfront.signers

```
class boto.cloudfront.signers.ActiveTrustedSigners
```

```

endElement (name, value, connection)
startElement (name, attrs, connection)

```

```
class boto.cloudfront.signers.Signer
```

```

endElement (name, value, connection)
startElement (name, attrs, connection)

```

```
class boto.cloudfront.signers.TrustedSigners
```

```

endElement (name, value, connection)
startElement (name, attrs, connection)

```

4.111.6 boto.cloudfront.invalidation

```
class boto.cloudfront.invalidation.InvalidationBatch (paths=None, connection=None, distribution=None, caller_reference="")
```

A simple invalidation request. :see: <http://docs.amazonwebservices.com/AmazonCloudFront/2010-08-01/APIReference/index.html?InvalidationBatchDatatype.html>

Create a new invalidation request: :paths: An array of paths to invalidate

```
add (path)
    Add another path to this invalidation request
```

```
endElement (name, value, connection)
```

```
escape (p)
    Escape a path, make sure it begins with a slash and contains no invalid characters. Retain literal wildcard characters.
```

```
remove (path)
    Remove a path from this invalidation request
```

```
startElement (name, attrs, connection)
```

```
to_xml ()
    Get this batch as XML
```

```
class boto.cloudfront.invalidation.InvalidationListResultSet (markers=None,  
connection=None,  
distribu-  
tion_id=None, in-  
validations=None,  
marker=","  
next_marker=None,  
max_items=None,  
is_truncated=False)
```

A resultset for listing invalidations on a given CloudFront distribution. Implements the iterator interface and transparently handles paging results from CF so even if you have many thousands of invalidations on the distribution you can iterate over all invalidations in a reasonably efficient manner.

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

```
to_boolean (value, true_value='true')
```

```
class boto.cloudfront.invalidation.InvalidationSummary (connection=None, distribu-  
tion_id=None, id="," sta-  
tus="")
```

Represents InvalidationSummary complex type in CloudFront API that lists the id and status of a given invalidation request.

```
endElement (name, value, connection)
```

```
get_distribution ()
```

Returns a Distribution object representing the parent CloudFront distribution of the invalidation request listed in the InvalidationSummary.

Return type *boto.cloudfront.distribution.Distribution*

Returns A Distribution object representing the parent CloudFront distribution of the invalidation request listed in the InvalidationSummary

```
get_invalidation_request ()
```

Returns an InvalidationBatch object representing the invalidation request referred to in the InvalidationSummary.

Return type *boto.cloudfront.invalidation.InvalidationBatch*

Returns An InvalidationBatch object representing the invalidation request referred to by the InvalidationSummary

```
startElement (name, attrs, connection)
```

4.111.7 boto.cloudfront.object

```
class boto.cloudfront.object.Object (bucket, name=None)
```

```
url (scheme='http')
```

```
class boto.cloudfront.object.StreamingObject (bucket, name=None)
```

```
url (scheme='rtmp')
```


4.111.8 boto.cloudfront.logging

class boto.cloudfront.logging.**LoggingInfo** (*bucket=""*, *prefix=""*)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

4.111.9 boto.cloudfront.exception

exception boto.cloudfront.exception.**CloudFrontServerError** (*status*, *reason*,
body=None, **args*)

4.112 CloudHSM

4.112.1 boto.cloudhsm

boto.cloudhsm.**connect_to_region** (*region_name*, ***kw_params*)

boto.cloudhsm.**regions** ()

Get all available regions for the AWS CloudHSM service.

Return type list

Returns A list of boto.regioninfo.RegionInfo

4.112.2 boto.cloudhsm.layer1

class boto.cloudhsm.layer1.**CloudHSMConnection** (***kwargs*)

AWS CloudHSM Service

APIVersion = '2014-05-30'

DefaultRegionEndpoint = 'cloudhsm.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.JSONResponseError*

ServiceName = 'CloudHSM'

TargetPrefix = 'CloudHsmFrontendService'

create_hapg (*label*)

Creates a high-availability partition group. A high-availability partition group is a group of partitions that spans multiple physical HSMs.

Parameters *label* (*string*) – The label of the new high-availability partition group.

create_hsm (*subnet_id*, *ssh_key*, *iam_role_arn*, *subscription_type*, *eni_ip=None*, *external_id=None*,
client_token=None, *syslog_ip=None*)

Creates an uninitialized HSM instance. Running this command provisions an HSM appliance and will result in charges to your AWS account for the HSM.

Parameters

- **subnet_id** (*string*) – The identifier of the subnet in your VPC in which to place the HSM.
- **ssh_key** (*string*) – The SSH public key to install on the HSM.
- **eni_ip** (*string*) – The IP address to assign to the HSM’s ENI.
- **iam_role_arn** (*string*) – The ARN of an IAM role to enable the AWS CloudHSM service to allocate an ENI on your behalf.
- **external_id** (*string*) – The external ID from **IamRoleArn**, if present.
- **subscription_type** (*string*) – The subscription type.
- **client_token** (*string*) – A user-defined token to ensure idempotence. Subsequent calls to this action with the same token will be ignored.
- **syslog_ip** (*string*) – The IP address for the syslog monitoring server.

create_luna_client (*certificate, label=None*)

Creates an HSM client.

Parameters

- **label** (*string*) – The label for the client.
- **certificate** (*string*) – The contents of a Base64-Encoded X.509 v3 certificate to be installed on the HSMs used by this client.

delete_hapg (*hapg_arn*)

Deletes a high-availability partition group.

Parameters hapg_arn (*string*) – The ARN of the high-availability partition group to delete.

delete_hsm (*hsm_arn*)

Deletes an HSM. Once complete, this operation cannot be undone and your key material cannot be recovered.

Parameters hsm_arn (*string*) – The ARN of the HSM to delete.

delete_luna_client (*client_arn*)

Deletes a client.

Parameters client_arn (*string*) – The ARN of the client to delete.

describe_hapg (*hapg_arn*)

Retrieves information about a high-availability partition group.

Parameters hapg_arn (*string*) – The ARN of the high-availability partition group to describe.

describe_hsm (*hsm_arn=None, hsm_serial_number=None*)

Retrieves information about an HSM. You can identify the HSM by its ARN or its serial number.

Parameters

- **hsm_arn** (*string*) – The ARN of the HSM. Either the **HsmArn** or the **SerialNumber** parameter must be specified.
- **hsm_serial_number** (*string*) – The serial number of the HSM. Either the **HsmArn** or the **HsmSerialNumber** parameter must be specified.

describe_luna_client (*client_arn=None, certificate_fingerprint=None*)

Retrieves information about an HSM client.

Parameters

- **client_arn** (*string*) – The ARN of the client.
- **certificate_fingerprint** (*string*) – The certificate fingerprint.

get_config (*client_arn, client_version, hapg_list*)

Gets the configuration files necessary to connect to all high availability partition groups the client is associated with.

Parameters

- **client_arn** (*string*) – The ARN of the client.
- **client_version** (*string*) – The client version.
- **hapg_list** (*list*) – A list of ARNs that identify the high-availability partition groups that are associated with the client.

list_available_zones ()

Lists the Availability Zones that have available AWS CloudHSM capacity.

list_hapgs (*next_token=None*)

Lists the high-availability partition groups for the account.

This operation supports pagination with the use of the NextToken member. If more results are available, the NextToken member of the response contains a token that you pass in the next call to ListHapgs to retrieve the next set of items.

Parameters next_token (*string*) – The NextToken value from a previous call to ListHapgs. Pass null if this is the first call.

list_hsms (*next_token=None*)

Retrieves the identifiers of all of the HSMs provisioned for the current customer.

This operation supports pagination with the use of the NextToken member. If more results are available, the NextToken member of the response contains a token that you pass in the next call to ListHsms to retrieve the next set of items.

Parameters next_token (*string*) – The NextToken value from a previous call to ListHsms. Pass null if this is the first call.

list_luna_clients (*next_token=None*)

Lists all of the clients.

This operation supports pagination with the use of the NextToken member. If more results are available, the NextToken member of the response contains a token that you pass in the next call to ListLunaClients to retrieve the next set of items.

Parameters next_token (*string*) – The NextToken value from a previous call to ListLunaClients. Pass null if this is the first call.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

modify_hapg (*hapg_arn, label=None, partition_serial_list=None*)

Modifies an existing high-availability partition group.

Parameters

- **hapg_arn** (*string*) – The ARN of the high-availability partition group to modify.
- **label** (*string*) – The new label for the high-availability partition group.

- **partition_serial_list** (*list*) – The list of partition serial numbers to make members of the high-availability partition group.

modify_hsm (*hsm_arn*, *subnet_id=None*, *eni_ip=None*, *iam_role_arn=None*, *external_id=None*, *syslog_ip=None*)
Modifies an HSM.

Parameters

- **hsm_arn** (*string*) – The ARN of the HSM to modify.
- **subnet_id** (*string*) – The new identifier of the subnet that the HSM is in.
- **eni_ip** (*string*) – The new IP address for the elastic network interface attached to the HSM.
- **iam_role_arn** (*string*) – The new IAM role ARN.
- **external_id** (*string*) – The new external ID.
- **syslog_ip** (*string*) – The new IP address for the syslog monitoring server.

modify_luna_client (*client_arn*, *certificate*)
Modifies the certificate used by the client.

This action can potentially start a workflow to install the new certificate on the client's HSMs.

Parameters

- **client_arn** (*string*) – The ARN of the client.
- **certificate** (*string*) – The new certificate for the client.

4.112.3 boto.cloudhsm.exceptions

exception boto.cloudhsm.exceptions.**CloudHsmInternalException** (*status*, *reason*,
body=None,
**args*)

exception boto.cloudhsm.exceptions.**CloudHsmServiceException** (*status*, *reason*,
body=None, **args*)

exception boto.cloudhsm.exceptions.**InvalidRequestException** (*status*, *reason*,
body=None, **args*)

4.113 Cloudsearch

4.113.1 boto.cloudsearch

boto.cloudsearch.**connect_to_region** (*region_name*, ***kw_params*)

boto.cloudsearch.**regions** ()

Get all available regions for the Amazon CloudSearch service.

Return type *list*

Returns A list of boto.regioninfo.RegionInfo

4.113.2 boto.cloudsearch.domain

class boto.cloudsearch.domain.**Domain** (*layer1*, *data*)

A Cloudsearch domain.

Variables

- **name** – The name of the domain.
- **id** – The internally generated unique identifier for the domain.
- **created** – A boolean which is True if the domain is created. It can take several minutes to initialize a domain when CreateDomain is called. Newly created search domains are returned with a False value for Created until domain creation is complete
- **deleted** – A boolean which is True if the search domain has been deleted. The system must clean up resources dedicated to the search domain when delete is called. Newly deleted search domains are returned from list_domains with a True value for deleted for several minutes until resource cleanup is complete.
- **processing** – True if processing is being done to activate the current domain configuration.
- **num_searchable_docs** – The number of documents that have been submitted to the domain and indexed.
- **requires_index_document** – True if index_documents needs to be called to activate the current domain configuration.
- **search_instance_count** – The number of search instances that are available to process search requests.
- **search_instance_type** – The instance type that is being used to process search requests.
- **search_partition_count** – The number of partitions across which the search index is spread.

create_index_field (*field_name*, *field_type*, *default*=”, *facet*=False, *result*=False, *searchable*=False, *source_attributes*=[])

Defines an IndexField, either replacing an existing definition or creating a new one.

Parameters

- **field_name** (*string*) – The name of a field in the search index.
- **field_type** (*string*) – The type of field. Valid values are `uint` | `literal` | `text`
- **default** (*string or int*) – The default value for the field. If the field is of type `uint` this should be an integer value. Otherwise, it’s a string.
- **facet** (*bool*) – A boolean to indicate whether facets are enabled for this field or not. Does not apply to fields of type `uint`.
- **results** (*bool*) – A boolean to indicate whether values of this field can be returned in search results or used in ranking. Does not apply to fields of type `uint`.
- **searchable** (*bool*) – A boolean to indicate whether search is enabled for this field or not. Applies only to fields of type `literal`.
- **source_attributes** (*list of dicts*) – An optional list of dicts that provide information about attributes for this index field. A maximum of 20 source attributes can be configured for each index field.

Each item in the list is a dict with the following keys:

- **data_copy** - The value is a dict with the following keys:
 - * **default** - Optional default value if the source attribute is not specified in a document.
 - * **name** - The name of the document source field to add to this `IndexField`.
- **data_function** - Identifies the transformation to apply when copying data from a source attribute.
- **data_map** - The value is a dict with the following keys:
 - * **cases** - A dict that translates source field values to custom values.
 - * **default** - An optional default value to use if the source attribute is not specified in a document.
 - * **name** - the name of the document source field to add to this `IndexField`
- **data_trim_title** - Trims common title words from a source document attribute when populating an `IndexField`. This can be used to create an `IndexField` you can use for sorting. The value is a dict with the following fields:
 - * **default** - An optional default value.
 - * **language** - an IETF RFC 4646 language code.
 - * **separator** - The separator that follows the text to trim.
 - * **name** - The name of the document source field to add.

Raises `BaseException`, `InternalException`, `LimitExceededException`, `InvalidTypeException`, `ResourceNotFoundException`

create_rank_expression (*name*, *expression*)

Create a new rank expression.

Parameters

- **rank_name** (*string*) - The name of an expression computed for ranking while processing a search request.
- **rank_expression** (*string*) - The expression to evaluate for ranking or thresholding while processing a search request. The `RankExpression` syntax is based on JavaScript expressions and supports:
 - Integer, floating point, hex and octal literals
 - **Shortcut evaluation of logical operators such that an** expression `a || b` evaluates to the value `a` if `a` is true without evaluating `b` at all
 - JavaScript order of precedence for operators
 - Arithmetic operators: `+` `-` `*` `/` `%`
 - Boolean operators (including the ternary operator)
 - Bitwise operators
 - Comparison operators
 - **Common mathematic functions:** `abs` `ceil` `erf` `exp` `floor` `lgamma` `ln` `log2` `log10` `max` `min` `sqrt` `pow`
 - **Trigonometric library functions:** `acosh` `acos` `asinh` `asin` `atanh` `atan` `cosh` `cos` `sinh` `sin` `tanh` `tan`
 - Random generation of a number between 0 and 1: `rand`
 - Current time in epoch: `time`
 - The min max functions that operate on a variable argument list

Intermediate results are calculated as double precision floating point values. The final return value of a RankExpression is automatically converted from floating point to a 32-bit unsigned integer by rounding to the nearest integer, with a natural floor of 0 and a ceiling of `max(uint32_t, 4294967295)`. Mathematical errors such as dividing by 0 will fail during evaluation and return a value of 0.

The source data for a RankExpression can be the name of an IndexField of type `uint`, another RankExpression or the reserved name `text_relevance`. The `text_relevance` source is defined to return an integer from 0 to 1000 (inclusive) to indicate how relevant a document is to the search request, taking into account repetition of search terms in the document and proximity of search terms to each other in each matching IndexField in the document.

For more information about using rank expressions to customize ranking, see the Amazon CloudSearch Developer Guide.

Raises BaseException, InternalException, LimitExceededException, InvalidTypeException, ResourceNotFoundException

created

delete ()

Delete this domain and all index data associated with it.

deleted

doc_service_arn

doc_service_endpoint

get_access_policies ()

Return a `boto.cloudsearch.option.OptionStatus` object representing the currently defined access policies for the domain.

get_document_service ()

get_index_fields (field_names=None)

Return a list of index fields defined for this domain.

get_rank_expressions (rank_names=None)

Return a list of rank expressions defined for this domain.

get_search_service ()

get_stemming ()

Return a `boto.cloudsearch.option.OptionStatus` object representing the currently defined stemming options for the domain.

get_stopwords ()

Return a `boto.cloudsearch.option.OptionStatus` object representing the currently defined stopword options for the domain.

get_synonyms ()

Return a `boto.cloudsearch.option.OptionStatus` object representing the currently defined synonym options for the domain.

id

index_documents ()

Tells the search domain to start indexing its documents using the latest text processing options and IndexFields. This operation must be invoked to make options whose OptionStatus has OptioState of RequiresIndexDocuments visible in search results.

```
name
num_searchable_docs
processing
requires_index_documents
search_instance_count
search_partition_count
search_service_arn
search_service_endpoint
update_from_data(data)
```

```
boto.cloudsearch.domain.handle_bool(value)
```

4.113.3 boto.cloudsearch.exceptions

4.113.4 boto.cloudsearch.layer1

```
class boto.cloudsearch.layer1.Layer1(aws_access_key_id=None,
                                     aws_secret_access_key=None,      is_secure=True,
                                     host=None,          port=None,      proxy=None,
                                     proxy_port=None,        proxy_user=None,
                                     proxy_pass=None,        debug=0,
                                     https_connection_factory=None,  region=None,
                                     path='/', api_version=None, security_token=None,
                                     validate_certs=True, profile_name=None)
```

```
APIVersion = '2011-02-01'
```

```
DefaultRegionEndpoint = 'cloudsearch.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
create_domain(domain_name)
```

Create a new search domain.

Parameters *domain_name* (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException, LimitExceededException

```
define_index_field(domain_name, field_name, field_type, default="", facet=False, result=False,
                  searchable=False, source_attributes=None)
```

Defines an IndexField, either replacing an existing definition or creating a new one.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **field_name** (*string*) – The name of a field in the search index.

- **field_type** (*string*) – The type of field. Valid values are `uint` | `literal` | `text`
- **default** (*string or int*) – The default value for the field. If the field is of type `uint` this should be an integer value. Otherwise, it's a string.
- **facet** (*bool*) – A boolean to indicate whether facets are enabled for this field or not. Does not apply to fields of type `uint`.
- **results** (*bool*) – A boolean to indicate whether values of this field can be returned in search results or used in ranking. Does not apply to fields of type `uint`.
- **searchable** (*bool*) – A boolean to indicate whether search is enabled for this field or not. Applies only to fields of type `literal`.
- **source_attributes** (*list of dicts*) – An optional list of dicts that provide information about attributes for this index field. A maximum of 20 source attributes can be configured for each index field.

Each item in the list is a dict with the following keys:

– **data_copy** - The value is a dict with the following keys:

- * **default** - Optional default value if the source attribute is not specified in a document.
- * **name** - The name of the document source field to add to this `IndexField`.

– **data_function** - Identifies the transformation to apply when copying data from a source attribute.

– **data_map** - The value is a dict with the following keys:

- * **cases** - A dict that translates source field values to custom values.
- * **default** - An optional default value to use if the source attribute is not specified in a document.
- * **name** - the name of the document source field to add to this `IndexField`

– **data_trim_title** - Trims common title words from a source document attribute when populating an `IndexField`. This can be used to create an `IndexField` you can use for sorting. The value is a dict with the following fields: * **default** - An optional default value. * **language** - an IETF RFC 4646 language code. * **separator** - The separator that follows the text to trim. * **name** - The name of the document source field to add.

Raises `BaseException`, `InternalException`, `LimitExceededException`, `InvalidTypeException`, `ResourceNotFoundException`

define_rank_expression (*domain_name, rank_name, rank_expression*)

Defines a `RankExpression`, either replacing an existing definition or creating a new one.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

- **rank_name** (*string*) – The name of an expression computed for ranking while processing a search request.
- **rank_expression** (*string*) – The expression to evaluate for ranking or thresholding while processing a search request. The RankExpression syntax is based on JavaScript expressions and supports:
 - Integer, floating point, hex and octal literals
 - **Shortcut evaluation of logical operators such that an** expression `a || b` evaluates to the value `a` if `a` is true without evaluating `b` at all
 - JavaScript order of precedence for operators
 - Arithmetic operators: `+` `-` `*` `/` `%`
 - Boolean operators (including the ternary operator)
 - Bitwise operators
 - Comparison operators
 - **Common mathematic functions:** `abs` `ceil` `erf` `exp` `floor` `lgamma` `ln` `log2` `log10` `max` `min` `sqrt` `pow`
 - **Trigonometric library functions:** `acosh` `acos` `asinh` `asin` `atanh` `atan` `cosh` `cos` `sinh` `sin` `tanh` `tan`
 - Random generation of a number between 0 and 1: `rand`
 - Current time in epoch: `time`
 - The min max functions that operate on a variable argument list

Intermediate results are calculated as double precision floating point values. The final return value of a RankExpression is automatically converted from floating point to a 32-bit unsigned integer by rounding to the nearest integer, with a natural floor of 0 and a ceiling of `max(uint32_t, 4294967295)`. Mathematical errors such as dividing by 0 will fail during evaluation and return a value of 0.

The source data for a RankExpression can be the name of an IndexField of type `uint`, another RankExpression or the reserved name `text_relevance`. The `text_relevance` source is defined to return an integer from 0 to 1000 (inclusive) to indicate how relevant a document is to the search request, taking into account repetition of search terms in the document and proximity of search terms to each other in each matching IndexField in the document.

For more information about using rank expressions to customize ranking, see the Amazon CloudSearch Developer Guide.

Raises BaseException, InternalException, LimitExceededException, InvalidTypeException, ResourceNotFoundException

delete_domain (*domain_name*)

Delete a search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException

delete_index_field (*domain_name*, *field_name*)

Deletes an existing `IndexField` from the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **field_name** (*string*) – A string that represents the name of an index field. Field names must begin with a letter and can contain the following characters: a-z (lowercase), 0-9, and _ (underscore). Uppercase letters and hyphens are not allowed. The names “body”, “docid”, and “text_relevance” are reserved and cannot be specified as field or rank expression names.

Raises `BaseException`, `InternalException`, `ResourceNotFoundException`

delete_rank_expression (*domain_name*, *rank_name*)

Deletes an existing `RankExpression` from the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **rank_name** (*string*) – Name of the `RankExpression` to delete.

Raises `BaseException`, `InternalException`, `ResourceNotFoundException`

describe_default_search_field (*domain_name*)

Describes options defining the default search field used by indexing for the search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises `BaseException`, `InternalException`, `ResourceNotFoundException`

describe_domains (*domain_names=None*)

Describes the domains (optionally limited to one or more domains by name) owned by this account.

Parameters **domain_names** (*list*) – Limits the response to the specified domains.

Raises `BaseException`, `InternalException`

describe_index_fields (*domain_name*, *field_names=None*)

Describes index fields in the search domain, optionally limited to a single `IndexField`.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **field_names** (*list*) – Limits the response to the specified fields.

Raises `BaseException`, `InternalException`, `ResourceNotFoundException`

describe_rank_expressions (*domain_name*, *rank_names=None*)

Describes RankExpressions in the search domain, optionally limited to a single expression.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **rank_names** (*list*) – Limit response to the specified rank names.

Raises BaseException, InternalException, ResourceNotFoundException

describe_service_access_policies (*domain_name*)

Describes the resource-based policies controlling access to the services in this search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException, ResourceNotFoundException

describe_stemming_options (*domain_name*)

Describes stemming options used by indexing for the search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException, ResourceNotFoundException

describe_stopword_options (*domain_name*)

Describes stopword options used by indexing for the search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException, ResourceNotFoundException

describe_synonym_options (*domain_name*)

Describes synonym options used by indexing for the search domain.

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises BaseException, InternalException, ResourceNotFoundException

get_response (*doc_path*, *action*, *params*, *path='/'*, *parent=None*, *verb='GET'*, *list_marker=None*)

index_documents (*domain_name*)

Tells the search domain to start scanning its documents using the latest text processing options and IndexFields. This operation must be invoked to make visible in searches any options whose `OptionStatus` has `OptionState` of `RequiresIndexDocuments`.

Parameters `domain_name` (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.

Raises `BaseException`, `InternalException`, `ResourceNotFoundException`

update_default_search_field (*domain_name*, *default_search_field*)

Updates options defining the default search field used by indexing for the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **default_search_field** (*string*) – The `IndexField` to use for search requests issued with the `q` parameter. The default is an empty string, which automatically searches all text fields.

Raises `BaseException`, `InternalException`, `InvalidTypeException`, `ResourceNotFoundException`

update_service_access_policies (*domain_name*, *access_policies*)

Updates the policies controlling access to the services in this search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **access_policies** (*string*) – An IAM access policy as described in The Access Policy Language in Using AWS Identity and Access Management. The maximum size of an access policy document is 100KB.

Raises `BaseException`, `InternalException`, `LimitExceededException`, `ResourceNotFoundException`, `InvalidTypeException`

update_stemming_options (*domain_name*, *stems*)

Updates stemming options used by indexing for the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **stems** (*string*) – Maps terms to their stems. The JSON object has a single key called “stems” whose value is a dict mapping terms to their stems. The maximum size of a stemming document is 500KB. Example: {“stems”:{“people”：“person”, “walking”：“walk”}}

Raises `BaseException`, `InternalException`, `InvalidTypeException`, `LimitExceededException`, `ResourceNotFoundException`

update_stopword_options (*domain_name*, *stopwords*)

Updates stopword options used by indexing for the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **stopwords** (*string*) – Lists stopwords in a JSON object. The object has a single key called “stopwords” whose value is an array of strings. The maximum size of a stopwords document is 10KB. Example: {“stopwords”: [“a”, “an”, “the”, “of”]}

Raises BaseException, InternalException, InvalidTypeException, LimitExceededException, ResourceNotFoundException

update_synonym_options (*domain_name*, *synonyms*)

Updates synonym options used by indexing for the search domain.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names must be unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen). Uppercase letters and underscores are not allowed.
- **synonyms** (*string*) – Maps terms to their synonyms. The JSON object has a single key “synonyms” whose value is a dict mapping terms to their synonyms. Each synonym is a simple string or an array of strings. The maximum size of a stopwords document is 100KB. Example: {“synonyms”: {“cat”: [“feline”, “kit-ten”], “puppy”: “dog”}}

Raises BaseException, InternalException, InvalidTypeException, LimitExceededException, ResourceNotFoundException

`boto.cloudsearch.layer1.do_bool` (*val*)

4.113.5 boto.cloudsearch.layer2

```
class boto.cloudsearch.layer2.Layer2 (aws_access_key_id=None,  
                                       aws_secret_access_key=None,      is_secure=True,  
                                       port=None,      proxy=None,      proxy_port=None,  
                                       host=None,      debug=0,      session_token=None, re-  
                                       gion=None, validate_certs=True)
```

create_domain (*domain_name*)

Create a new CloudSearch domain and return the corresponding `boto.cloudsearch.domain.Domain` object.

list_domains (*domain_names=None*)

Return a list of `boto.cloudsearch.domain.Domain` objects for each domain defined in the current account.

lookup (*domain_name*)

Lookup a single domain :param domain_name: The name of the domain to look up :type domain_name: str

Returns Domain object, or None if the domain isn't found

Return type `boto.cloudsearch.domain.Domain`

4.113.6 boto.cloudsearch.optionstatus

```
class boto.cloudsearch.optionstatus.IndexFieldStatus (domain,          data=None,
                                                    refresh_fn=None,
                                                    save_fn=None)
```

save ()

Write the current state of the local object back to the CloudSearch service.

```
class boto.cloudsearch.optionstatus.OptionStatus (domain,          data=None,          re-
                                                    fresh_fn=None, save_fn=None)
```

Presents a combination of status field (defined below) which are accessed as attributes and option values which are stored in the native Python dictionary. In this class, the option values are merged from a JSON object that is stored as the Option part of the object.

Variables

- **domain_name** – The name of the domain this option is associated with.
- **create_date** – A timestamp for when this option was created.
- **state** – The state of processing a change to an option. Possible values:
 - RequiresIndexDocuments: the option's latest value will not be visible in searches until IndexDocuments has been called and indexing is complete.
 - Processing: the option's latest value is not yet visible in all searches but is in the process of being activated.
 - Active: the option's latest value is completely visible.
- **update_date** – A timestamp for when this option was updated.
- **update_version** – A unique integer that indicates when this option was last updated.

endElement (*name, value, connection*)

refresh (*data=None*)

Refresh the local state of the object. You can either pass new state data in as the parameter `data` or, if that parameter is omitted, the state data will be retrieved from CloudSearch.

save ()

Write the current state of the local object back to the CloudSearch service.

startElement (*name, attrs, connection*)

to_json ()

Return the JSON representation of the options as a string.

wait_for_state (*state*)

Performs polling of CloudSearch to wait for the `state` of this object to change to the provided state.

```
class boto.cloudsearch.optionstatus.RankExpressionStatus (domain,          data=None,
                                                    refresh_fn=None,
                                                    save_fn=None)
```

```
class boto.cloudsearch.optionstatus.ServicePoliciesStatus (domain,          data=None,
                                                    refresh_fn=None,
                                                    save_fn=None)
```

allow_doc_ip (*ip*)

Add the provided ip address or CIDR block to the list of allowable address for the document service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

allow_search_ip (*ip*)

Add the provided ip address or CIDR block to the list of allowable address for the search service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

disallow_doc_ip (*ip*)

Remove the provided ip address or CIDR block from the list of allowable address for the document service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

disallow_search_ip (*ip*)

Remove the provided ip address or CIDR block from the list of allowable address for the search service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

new_statement (*arn, ip*)

Returns a new policy statement that will allow access to the service described by *arn* by the ip specified in *ip*.

Parameters

- **arn** (*string*) – The Amazon Resource Notation identifier for the service you wish to provide access to. This would be either the search service or the document service.
- **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

4.113.7 boto.cloudsearch.search

exception boto.cloudsearch.search.**CommitMismatchError**

class boto.cloudsearch.search.**Query** (*q=None, bq=None, rank=None, return_fields=None, size=10, start=0, facet=None, facet_constraints=None, facet_sort=None, facet_top_n=None, t=None*)

RESULTS_PER_PAGE = 500

to_params ()

Transform search parameters from instance properties to a dictionary

Return type dict

Returns search parameters

update_size (*new_size*)

class boto.cloudsearch.search.**SearchConnection** (*domain=None, endpoint=None*)

build_query (*q=None, bq=None, rank=None, return_fields=None, size=10, start=0, facet=None, facet_constraints=None, facet_sort=None, facet_top_n=None, t=None*)

get_all_hits (*query*)

Get a generator to iterate over all search results

Transparently handles the results paging from Cloudsearch search results so even if you have many thousands of results you can iterate over all results in a reasonably efficient manner.

Parameters **query** (*boto.cloudsearch.search.Query*) – A group of search criteria

Return type generator

Returns All docs matching query

get_all_paged (*query*, *per_page*)

Get a generator to iterate over all pages of search results

Parameters

- **query** (*boto.cloudsearch.search.Query*) – A group of search criteria
- **per_page** (*int*) – Number of docs in each *boto.cloudsearch.search.SearchResults* object.

Return type generator

Returns Generator containing *boto.cloudsearch.search.SearchResults*

get_num_hits (*query*)

Return the total number of hits for query

Parameters **query** (*boto.cloudsearch.search.Query*) – a group of search criteria

Return type *int*

Returns Total number of hits for query

search (*q=None*, *bq=None*, *rank=None*, *return_fields=None*, *size=10*, *start=0*, *facet=None*, *facet_constraints=None*, *facet_sort=None*, *facet_top_n=None*, *t=None*)

Send a query to CloudSearch

Each search query should use at least the *q* or *bq* argument to specify the search parameter. The other options are used to specify the criteria of the search.

Parameters

- **q** (*string*) – A string to search the default search fields for.
- **bq** (*string*) – A string to perform a Boolean search. This can be used to create advanced searches.
- **rank** (*List of strings*) – A list of fields or rank expressions used to order the search results. A field can be reversed by using the *-* operator. `['-year', 'author']`
- **return_fields** (*List of strings*) – A list of fields which should be returned by the search. If this field is not specified, only IDs will be returned. `['headline']`
- **size** (*int*) – Number of search results to specify
- **start** (*int*) – Offset of the first search result to return (can be used for paging)
- **facet** (*list*) – List of fields for which facets should be returned `['colour', 'size']`
- **facet_constraints** (*dict*) – Use to limit facets to specific values specified as comma-delimited strings in a Dictionary of facets `{'colour': "blue,white,red", 'size': "big"}`
- **facet_sort** (*dict*) – Rules used to specify the order in which facet values should be returned. Allowed values are *alpha*, *count*, *max*, *sum*. Use *alpha* to sort alphabetical, and *count* to sort the facet by number of available result. `{'color': 'alpha', 'size': 'count'}`
- **facet_top_n** (*dict*) – Dictionary of facets and number of facets to return. `{'colour': 2}`
- **t** (*dict*) – Specify ranges for specific fields `{'year': '2000..2005'}`

Return type *boto.cloudsearch.search.SearchResults*

Returns Returns the results of this search

The following examples all assume we have indexed a set of documents with fields: *author, date, headline*

A simple search will look for documents whose default text search fields will contain the search word exactly:

```
>>> search(q='Tim') # Return documents with the word Tim in them (but not_
↳ Timothy)
```

A simple search with more keywords will return documents whose default text search fields contain the search strings together or separately.

```
>>> search(q='Tim apple') # Will match "tim" and "apple"
```

More complex searches require the boolean search operator.

Wildcard searches can be used to search for any words that start with the search string.

```
>>> search(bq="'Tim*') # Return documents with words like Tim or Timothy)
```

Search terms can also be combined. Allowed operators are “and”, “or”, “not”, “field”, “optional”, “token”, “phrase”, or “filter”

```
>>> search(bq="(and 'Tim' (field author 'John Smith'))")
```

Facets allow you to show classification information about the search results. For example, you can retrieve the authors who have written about Tim:

```
>>> search(q='Tim', facet=['Author'])
```

With `facet_constraints`, `facet_top_n` and `facet_sort` more complicated constraints can be specified such as returning the top author out of John Smith and Mark Smith who have a document with the word Tim in it.

```
>>> search(q='Tim',
...       facet=['Author'],
...       facet_constraints={'author': "'John Smith','Mark Smith'"},
...       facet=['author'],
...       facet_top_n={'author': 1},
...       facet_sort={'author': 'count'})
```

class `boto.cloudsearch.search.SearchResults` (***attrs*)

next_page ()

Call Cloudsearch to get the next page of search results

Return type *boto.cloudsearch.search.SearchResults*

Returns the following page of search results

exception `boto.cloudsearch.search.SearchServiceException`

4.113.8 boto.cloudsearch.document

exception `boto.cloudsearch.document.CommitMismatchError`

class `boto.cloudsearch.document.CommitResponse` (*response, doc_service, sdf*)

Wrapper for response to Cloudsearch document batch commit.

Parameters

- **response** (`requests.models.Response`) – Response from Cloudsearch /documents/batch API
- **doc_service** (`boto.cloudsearch.document.DocumentServiceConnection`) – Object containing the documents posted and methods to retry

Raises `boto.exception.BotoServerError`

Raises `boto.cloudsearch.document.SearchServiceException`

Raises `boto.cloudsearch.document.EncodingError`

Raises `boto.cloudsearch.document.ContentTooLongError`

exception `boto.cloudsearch.document.ContentTooLongError`

Content sent for Cloud Search indexing was too long

This will usually happen when documents queued for indexing add up to more than the limit allowed per upload batch (5MB)

class `boto.cloudsearch.document.DocumentServiceConnection` (*domain=None, endpoint=None*)

A CloudSearch document service.

The DocumentServiceConnection is used to add, remove and update documents in CloudSearch. Commands are uploaded to CloudSearch in SDF (Search Document Format).

To generate an appropriate SDF, use `add()` to add or update documents, as well as `delete()` to remove documents.

Once the set of documents is ready to be index, use `commit()` to send the commands to CloudSearch.

If there are a lot of documents to index, it may be preferable to split the generation of SDF data and the actual uploading into CloudSearch. Retrieve the current SDF with `get_sdf()`. If this file is the uploaded into S3, it can be retrieved back afterwards for upload into CloudSearch using `add_sdf_from_s3()`.

The SDF is not cleared after a `commit()`. If you wish to continue using the DocumentServiceConnection for another batch upload of commands, you will need to `clear_sdf()` first to stop the previous batch of commands from being uploaded again.

add (*_id, version, fields, lang='en'*)

Add a document to be processed by the DocumentService

The document will not actually be added until `commit()` is called

Parameters

- **_id** (*string*) – A unique ID used to refer to this document.
- **version** (*int*) – Version of the document being indexed. If a file is being reindexed, the version should be higher than the existing one in CloudSearch.
- **fields** (*dict*) – A dictionary of key-value pairs to be uploaded .
- **lang** (*string*) – The language code the data is in. Only 'en' is currently supported

add_sdf_from_s3 (*key_obj*)

Load an SDF from S3

Using this method will result in documents added through `add()` and `delete()` being ignored.

Parameters **key_obj** (`boto.s3.key.Key`) – An S3 key which contains an SDF

clear_sdf ()

Clear the working documents from this DocumentServiceConnection

This should be used after `commit()` if the connection will be reused for another set of documents.

commit()

Actually send an SDF to CloudSearch for processing

If an SDF file has been explicitly loaded it will be used. Otherwise, documents added through `add()` and `delete()` will be used.

Return type `CommitResponse`

Returns A summary of documents added and deleted

delete(_id, version)

Schedule a document to be removed from the CloudSearch service

The document will not actually be scheduled for removal until `commit()` is called

Parameters

- **_id** (`string`) – The unique ID of this document.
- **version** (`int`) – Version of the document to remove. The delete will only occur if this version number is higher than the version currently in the index.

get_sdf()

Generate the working set of documents in Search Data Format (SDF)

Return type `string`

Returns JSON-formatted string of the documents in SDF

exception `boto.cloudsearch.document.EncodingError`

Content sent for Cloud Search indexing was incorrectly encoded.

This usually happens when a document is marked as unicode but non-unicode characters are present.

exception `boto.cloudsearch.document.SearchServiceException`

4.114 Cloudsearch

4.114.1 boto.cloudsearch2

`boto.cloudsearch2.connect_to_region(region_name, **kw_params)`

`boto.cloudsearch2.regions()`

Get all available regions for the Amazon CloudSearch service.

Return type `list`

Returns A list of `boto.regioninfo.RegionInfo`

4.114.2 boto.cloudsearch2.domain

class `boto.cloudsearch2.domain.Domain(layer1, data)`

A Cloudsearch domain.

Variables

- **name** – The name of the domain.
- **id** – The internally generated unique identifier for the domain.
- **created** – A boolean which is True if the domain is created. It can take several minutes to initialize a domain when `CreateDomain` is called. Newly created search domains are returned with a False value for `Created` until domain creation is complete

- ***deleted*** – A boolean which is True if the search domain has been deleted. The system must clean up resources dedicated to the search domain when delete is called. Newly deleted search domains are returned from list_domains with a True value for deleted for several minutes until resource cleanup is complete.
- ***processing*** – True if processing is being done to activate the current domain configuration.
- ***num_searchable_docs*** – The number of documents that have been submitted to the domain and indexed.
- ***requires_index_document*** – True if index_documents needs to be called to activate the current domain configuration.
- ***search_instance_count*** – The number of search instances that are available to process search requests.
- ***search_instance_type*** – The instance type that is being used to process search requests.
- ***search_partition_count*** – The number of partitions across which the search index is spread.

Constructor - Create a domain object from a layer1 and data params

Parameters **layer1** (boto.cloudsearch2.layer1.Layer1 object) – A boto.cloudsearch2.layer1.Layer1 object which is used to perform operations on the domain.

create_expression (*name, value*)

Create a new expression.

Parameters

- **name** (*string*) – The name of an expression for processing during a search request.
- **value** (*string*) – The expression to evaluate for ranking or thresholding while processing a search request. The Expression syntax is based on JavaScript expressions and supports:
 - Single value, sort enabled numeric fields (int, double, date)
 - Other expressions
 - The `_score` variable, which references a document's relevance score
 - The `_time` variable, which references the current epoch time
 - Integer, floating point, hex, and octal literals
 - Arithmetic operators: `+ - * / %`
 - Bitwise operators: `| & ^ ~ << >> >>>`
 - Boolean operators (including the ternary operator): `&& || ! ?:`
 - Comparison operators: `< <= == >= >`
 - Mathematical functions: `abs ceil exp floor ln log2 log10 logn`
`max min pow sqrt pow`
 - Trigonometric functions: `acos acosh asin asinh atan atan2 atanh`
`cos cosh sin sinh tanh tan`

- The haversin distance function

Expressions always return an integer value from 0 to the maximum 64-bit signed integer value ($2^{63} - 1$). Intermediate results are calculated as double-precision floating point values and the return value is rounded to the nearest integer. If the expression is invalid or evaluates to a negative value, it returns 0. If the expression evaluates to a value greater than the maximum, it returns the maximum value.

The source data for an Expression can be the name of an IndexField of type `int` or `double`, another Expression or the reserved name `_score`. The `_score` source is defined to return as a double from 0 to 10.0 (inclusive) to indicate how relevant a document is to the search request, taking into account repetition of search terms in the document and proximity of search terms to each other in each matching IndexField in the document.

For more information about using rank expressions to customize ranking, see the Amazon CloudSearch Developer Guide.

Returns ExpressionStatus object

Return type `boto.cloudsearch2.option.ExpressionStatus` object

Raises BaseException, InternalException, LimitExceededException, InvalidTypeException, ResourceNotFoundException

create_index_field (*field_name*, *field_type*, *default=""*, *facet=False*, *returnable=False*, *searchable=False*, *sortable=False*, *highlight=False*, *source_field=None*, *analysis_scheme=None*)

Defines an IndexField, either replacing an existing definition or creating a new one.

Parameters

- **field_name** (*string*) – The name of a field in the search index.
- **field_type** (*string*) – The type of field. Valid values are `int` | `double` | `literal` | `text` | `date` | `latlon` | `int-array` | `double-array` | `literal-array` | `text-array` | `date-array`
- **default** (*string or int*) – The default value for the field. If the field is of type `int` this should be an integer value. Otherwise, it's a string.
- **facet** (*bool*) – A boolean to indicate whether facets are enabled for this field or not. Does not apply to fields of type `int`, `int-array`, `text`, `text-array`.
- **returnable** (*bool*) – A boolean to indicate whether values of this field can be returned in search results or used in ranking.
- **searchable** (*bool*) – A boolean to indicate whether search is enabled for this field or not.
- **sortable** (*bool*) – A boolean to indicate whether sorting is enabled for this field or not. Does not apply to fields of array types.
- **highlight** (*bool*) – A boolean to indicate whether highlighting is enabled for this field or not. Does not apply to fields of type `double`, `int`, `date`, `latlon`
- **source_field** (*list of strings or string*) – For array types, this is the list of fields to treat as the source. For singular types, pass a string only.
- **analysis_scheme** (*string*) – The analysis scheme to use for this field. Only applies to `text` | `text-array` field types

Returns IndexFieldStatus objects

Return type boto.cloudsearch2.option.IndexFieldStatus object

Raises BaseException, InternalException, LimitExceededException, InvalidTypeException, ResourceNotFoundException

created

delete()

Delete this domain and all index data associated with it.

deleted

doc_service_endpoint

get_access_policies()

Return a boto.cloudsearch2.option.ServicePoliciesStatus object representing the currently defined access policies for the domain. :return: ServicePoliciesStatus object :rtype: boto.cloudsearch2.option.ServicePoliciesStatus object

get_analysis_schemes()

Return a list of Analysis Scheme objects.

get_availability_options()

Return a boto.cloudsearch2.option.AvailabilityOptionsStatus object representing the currently defined availability options for the domain. :return: OptionsStatus object :rtype: boto.cloudsearch2.option.AvailabilityOptionsStatus object

get_document_service()

get_expressions (*names=None*)

Return a list of rank expressions defined for this domain. :return: list of ExpressionStatus objects :rtype: list of boto.cloudsearch2.option.ExpressionStatus object

get_index_fields (*field_names=None*)

Return a list of index fields defined for this domain. :return: list of IndexFieldStatus objects :rtype: list of boto.cloudsearch2.option.IndexFieldStatus object

get_scaling_options()

Return a boto.cloudsearch2.option.ScalingParametersStatus object representing the currently defined scaling options for the domain. :return: ScalingParametersStatus object :rtype: boto.cloudsearch2.option.ScalingParametersStatus object

get_search_service()

id

index_documents()

Tells the search domain to start indexing its documents using the latest text processing options and IndexFields. This operation must be invoked to make options whose OptionStatus has OptionState of RequiresIndexDocuments visible in search results.

name

processing

requires_index_documents

search_instance_count

search_partition_count

`search_service_endpoint`

`service_arn`

`update_from_data` (*data*)

`boto.cloudsearch2.domain.handle_bool` (*value*)

4.114.3 boto.cloudsearch2.layer1

class `boto.cloudsearch2.layer1.CloudSearchConnection` (***kwargs*)

Amazon CloudSearch Configuration Service You use the Amazon CloudSearch configuration service to create, configure, and manage search domains. Configuration service requests are submitted using the AWS Query protocol. AWS Query requests are HTTP or HTTPS requests submitted via HTTP GET or POST with a query parameter named Action.

The endpoint for configuration service requests is region- specific: `cloudsearch.region.amazonaws.com`. For example, `cloudsearch.us-east-1.amazonaws.com`. For a current list of supported regions and endpoints, see [‘Regions and Endpoints’](#).

`APIVersion` = '2013-01-01'

`DefaultRegionEndpoint` = 'cloudsearch.us-east-1.amazonaws.com'

`DefaultRegionName` = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

build_complex_param (*params*, *label*, *value*)

Serialize a structure.

For example:

```
param_type = 'structure'
label = 'IndexField'
value = {'IndexFieldName': 'a', 'IntOptions': {'DefaultValue': 5}}
```

would result in the params dict being updated with these params:

```
IndexField.IndexFieldName = a
IndexField.IntOptions.DefaultValue = 5
```

Parameters

- **params** (*dict*) – The params dict. The complex list params will be added to this dict.
- **label** (*str*) – String label for param key
- **value** (*any*) – The value to serialize

build_suggesters (*domain_name*)

Indexes the search suggestions.

Parameters `domain_name` (*string*) – A string that represents the name of a domain.

Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

create_domain (*domain_name*)

Creates a new search domain. For more information, see [‘Creating a Search Domain’](#) in the Amazon CloudSearch Developer Guide .

Parameters **domain_name** (*string*) – A name for the domain you are creating. Allowed characters are a-z (lower-case letters), 0-9, and hyphen (-). Domain names must start with a letter or number and be at least 3 and no more than 28 characters long.

define_analysis_scheme (*domain_name, analysis_scheme*)

Configures an analysis scheme that can be applied to a *text* or *text-array* field to define language-specific text processing options. For more information, see [‘Configuring Analysis Schemes’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **analysis_scheme** (*dict*) – Configuration information for an analysis scheme. Each analysis scheme has a unique name and specifies the language of the text to be processed. The following options can be configured for an analysis scheme: *Synonyms*, *Stopwords*, *StemmingDictionary*, and *AlgorithmicStemming*.

define_expression (*domain_name, expression*)

Configures an *Expression* for the search domain. Used to create new expressions and modify existing ones. If the expression exists, the new configuration replaces the old one. For more information, see [‘Configuring Expressions’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **expression** (*dict*) – A named expression that can be evaluated at search time. Can be used to sort the search results, define other expressions, or return computed information in the search results.

define_index_field (*domain_name, index_field*)

Configures an *IndexField* for the search domain. Used to create new fields and modify existing ones. You must specify the name of the domain you are configuring and an index field configuration. The index field configuration specifies a unique name, the index field type, and the options you want to configure for the field. The options you can specify depend on the *IndexFieldType*. If the field exists, the new configuration replaces the old one. For more information, see [‘Configuring Index Fields’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **index_field** (*dict*) – The index field and field options you want to configure.

define_suggester (*domain_name, suggester*)

Configures a suggester for a domain. A suggester enables you to display possible matches before users finish typing their queries. When you configure a suggester, you must specify the name of the text field

you want to search for possible matches and a unique name for the suggester. For more information, see [‘Getting Search Suggestions’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **suggester** (*dict*) – Configuration information for a search suggester. Each suggester has a unique name and specifies the text field you want to use for suggestions. The following options can be configured for a suggester: *FuzzyMatching*, *SortExpression*.

delete_analysis_scheme (*domain_name*, *analysis_scheme_name*)

Deletes an analysis scheme. For more information, see [‘Configuring Analysis Schemes’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **analysis_scheme_name** (*string*) – The name of the analysis scheme you want to delete.

delete_domain (*domain_name*)

Permanently deletes a search domain and all of its data. Once a domain has been deleted, it cannot be recovered. For more information, see [‘Deleting a Search Domain’](#) in the Amazon CloudSearch Developer Guide .

Parameters **domain_name** (*string*) – The name of the domain you want to permanently delete.

delete_expression (*domain_name*, *expression_name*)

Removes an *Expression* from the search domain. For more information, see [‘Configuring Expressions’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **expression_name** (*string*) – The name of the *Expression* to delete.

delete_index_field (*domain_name*, *index_field_name*)

Removes an *IndexField* from the search domain. For more information, see [‘Configuring Index Fields’](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **index_field_name** (*string*) – The name of the index field your want to remove from the domain’s indexing options.

delete_suggester (*domain_name*, *suggester_name*)

Deletes a suggester. For more information, see [‘Getting Search Suggestions’_](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **suggester_name** (*string*) – Specifies the name of the suggester you want to delete.

describe_analysis_schemes (*domain_name*, *analysis_scheme_names=None*, *deployed=None*)

Gets the analysis schemes configured for a domain. An analysis scheme defines language-specific text processing options for a *text* field. Can be limited to specific analysis schemes by name. By default, shows all analysis schemes and includes any pending changes to the configuration. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see [‘Configuring Analysis Schemes’_](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.
- **analysis_scheme_names** (*list*) – The analysis schemes you want to describe.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

describe_availability_options (*domain_name*, *deployed=None*)

Gets the availability options configured for a domain. By default, shows the configuration with any pending changes. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see [‘Configuring Availability Options’_](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

describe_domains (*domain_names=None*)

Gets information about the search domains owned by this account. Can be limited to specific domains. Shows all domains by default. To get the number of searchable documents in a domain, use the console or submit a *matchall* request to your domain’s search endpoint: *q=matchall&q.parser=structured&size=0*. For more information, see [‘Getting Information about a Search Domain’_](#) in the Amazon CloudSearch Developer Guide .

Parameters **domain_names** (*list*) – The names of the domains you want to include in the response.

describe_expressions (*domain_name*, *expression_names=None*, *deployed=None*)

Gets the expressions configured for the search domain. Can be limited to specific expressions by name. By default, shows all expressions and includes any pending changes to the configuration. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see [‘Configuring Expressions’_](#) in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.

- **expression_names** (*list*) – Limits the *DescribeExpressions* response to the specified expressions. If not specified, all expressions are shown.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

describe_index_fields (*domain_name, field_names=None, deployed=None*)

Gets information about the index fields configured for the search domain. Can be limited to specific fields by name. By default, shows all fields and includes any pending changes to the configuration. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see **‘Getting Domain Information’** in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.
- **field_names** (*list*) – A list of the index fields you want to describe. If not specified, information is returned for all configured index fields.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

describe_scaling_parameters (*domain_name*)

Gets the scaling parameters configured for a domain. A domain’s scaling parameters specify the desired search instance type and replication count. For more information, see **‘Configuring Scaling Options’** in the Amazon CloudSearch Developer Guide .

Parameters **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

describe_service_access_policies (*domain_name, deployed=None*)

Gets information about the access policies that control access to the domain’s document and search endpoints. By default, shows the configuration with any pending changes. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see **‘Configuring Access for a Search Domain’** in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

describe_suggesters (*domain_name, suggester_names=None, deployed=None*)

Gets the suggesters configured for a domain. A suggester enables you to display possible matches before users finish typing their queries. Can be limited to specific suggesters by name. By default, shows all suggesters and includes any pending changes to the configuration. Set the *Deployed* option to *True* to show the active configuration and exclude pending changes. For more information, see **‘Getting Search Suggestions’** in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – The name of the domain you want to describe.
- **suggester_names** (*list*) – The suggesters you want to describe.
- **deployed** (*boolean*) – Whether to display the deployed configuration (*True*) or include any pending changes (*False*). Defaults to *False*.

index_documents (*domain_name*)

Tells the search domain to start indexing its documents using the latest indexing options. This operation must be invoked to activate options whose *OptionStatus* is *RequiresIndexDocuments*.

Parameters `domain_name` (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

list_domain_names ()

Lists all search domains owned by an account.

update_availability_options (*domain_name*, *multi_az*)

Configures the availability options for a domain. Enabling the Multi-AZ option expands an Amazon CloudSearch domain to an additional Availability Zone in the same Region to increase fault tolerance in the event of a service disruption. Changes to the Multi-AZ option can take about half an hour to become active. For more information, see ‘**Configuring Availability Options**’_ in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **multi_az** (*boolean*) – You expand an existing search domain to a second Availability Zone by setting the Multi-AZ option to true. Similarly, you can turn off the Multi-AZ option to downgrade the domain to a single Availability Zone by setting the Multi-AZ option to *False*.

update_scaling_parameters (*domain_name*, *scaling_parameters*)

Configures scaling parameters for a domain. A domain’s scaling parameters specify the desired search instance type and replication count. Amazon CloudSearch will still automatically scale your domain based on the volume of data and traffic, but not below the desired instance type and replication count. If the Multi-AZ option is enabled, these values control the resources used per Availability Zone. For more information, see ‘**Configuring Scaling Options**’_ in the Amazon CloudSearch Developer Guide .

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **scaling_parameters** (*dict*) – The desired instance type and desired number of replicas of each index partition.

update_service_access_policies (*domain_name*, *access_policies*)

Configures the access rules that control access to the domain’s document and search endpoints. For more information, see ‘**Configuring Access for an Amazon CloudSearch Domain**’_.

Parameters

- **domain_name** (*string*) – A string that represents the name of a domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **access_policies** (*string*) – The access rules you want to configure. These rules replace any existing rules.

4.114.4 boto.cloudsearch2.layer2

```
class boto.cloudsearch2.layer2.Layer2 (aws_access_key_id=None,
                                         aws_secret_access_key=None,      is_secure=True,
                                         port=None,      proxy=None,      proxy_port=None,
                                         host=None,      debug=0,      session_token=None,
                                         region=None,      validate_certs=True,
                                         sign_request=False)
```

create_domain (*domain_name*)

Create a new CloudSearch domain and return the corresponding object. :return: Domain object, or None if the domain isn't found :rtype: *boto.cloudsearch2.domain.Domain*

list_domains (*domain_names=None*)

Return a list of objects for each domain defined in the current account. :rtype: list of *boto.cloudsearch2.domain.Domain*

lookup (*domain_name*)

Lookup a single domain :param domain_name: The name of the domain to look up :type domain_name: str

Returns Domain object, or None if the domain isn't found

Return type *boto.cloudsearch2.domain.Domain*

4.114.5 boto.cloudsearch2.optionstatus

```
class boto.cloudsearch2.optionstatus.AvailabilityOptionsStatus (domain,
                                                                data=None, re-
                                                                fresh_fn=None,
                                                                re-
                                                                fresh_key=None,
                                                                save_fn=None)
```

save ()

Write the current state of the local object back to the CloudSearch service.

```
class boto.cloudsearch2.optionstatus.ExpressionStatus (domain,      data=None,
                                                         refresh_fn=None,      re-
                                                         fresh_key=None,
                                                         save_fn=None)
```

```
class boto.cloudsearch2.optionstatus.IndexFieldStatus (domain,      data=None,
                                                         refresh_fn=None,      re-
                                                         fresh_key=None,
                                                         save_fn=None)
```

save ()

Write the current state of the local object back to the CloudSearch service.

```
class boto.cloudsearch2.optionstatus.OptionStatus (domain,      data=None, re-
                                                         fresh_fn=None, refresh_key=None,
                                                         save_fn=None)
```

Presents a combination of status field (defined below) which are accessed as attributes and option values which are stored in the native Python dictionary. In this class, the option values are merged from a JSON object that is stored as the Option part of the object.

Variables

- **domain_name** – The name of the domain this option is associated with.

- **create_date** – A timestamp for when this option was created.
- **state** – The state of processing a change to an option. Possible values:
 - **RequiresIndexDocuments**: the option’s latest value will not be visible in searches until `IndexDocuments` has been called and indexing is complete.
 - **Processing**: the option’s latest value is not yet visible in all searches but is in the process of being activated.
 - **Active**: the option’s latest value is completely visible.
- **update_date** – A timestamp for when this option was updated.
- **update_version** – A unique integer that indicates when this option was last updated.

refresh (*data=None*)

Refresh the local state of the object. You can either pass new state data in as the parameter `data` or, if that parameter is omitted, the state data will be retrieved from `CloudSearch`.

save ()

Write the current state of the local object back to the `CloudSearch` service.

to_json ()

Return the JSON representation of the options as a string.

```
class boto.cloudsearch2.optionstatus.ScalingParametersStatus (domain,
                                                    data=None, re-
                                                    fresh_fn=None,
                                                    refresh_key=None,
                                                    save_fn=None)
```

```
class boto.cloudsearch2.optionstatus.ServicePoliciesStatus (domain, data=None,
                                                    refresh_fn=None,
                                                    refresh_key=None,
                                                    save_fn=None)
```

allow_doc_ip (*ip*)

Add the provided ip address or CIDR block to the list of allowable address for the document service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

allow_search_ip (*ip*)

Add the provided ip address or CIDR block to the list of allowable address for the search service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

disallow_doc_ip (*ip*)

Remove the provided ip address or CIDR block from the list of allowable address for the document service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

disallow_search_ip (*ip*)

Remove the provided ip address or CIDR block from the list of allowable address for the search service.

Parameters **ip** (*string*) – An IP address or CIDR block you wish to grant access to.

new_statement (*arn, ip*)

Returns a new policy statement that will allow access to the service described by `arn` by the ip specified in `ip`.

Parameters

- **arn** (*string*) – The Amazon Resource Notation identifier for the service you wish to provide access to. This would be either the search service or the document service.

- `ip` (*string*) – An IP address or CIDR block you wish to grant access to.

4.114.6 boto.cloudsearch2.search

```
class boto.cloudsearch2.search.Query (q=None, parser=None, fq=None, expr=None, return_fields=None, size=10, start=0, sort=None, facet=None, highlight=None, partial=None, options=None)
```

```
RESULTS_PER_PAGE = 500
```

```
to_domain_connection_params ()
```

Transform search parameters from instance properties to a dictionary that CloudSearchDomainConnection can accept

Return type dict

Returns search parameters

```
to_params ()
```

Transform search parameters from instance properties to a dictionary

Return type dict

Returns search parameters

```
update_size (new_size)
```

```
class boto.cloudsearch2.search.SearchConnection (domain=None, endpoint=None)
```

```
build_query (q=None, parser=None, fq=None, rank=None, return_fields=None, size=10, start=0, facet=None, highlight=None, sort=None, partial=None, options=None)
```

```
get_all_hits (query)
```

Get a generator to iterate over all search results

Transparently handles the results paging from Cloudsearch search results so even if you have many thousands of results you can iterate over all results in a reasonably efficient manner.

Parameters `query` (*boto.cloudsearch2.search.Query*) – A group of search criteria

Return type generator

Returns All docs matching query

```
get_all_paged (query, per_page)
```

Get a generator to iterate over all pages of search results

Parameters

- `query` (*boto.cloudsearch2.search.Query*) – A group of search criteria

- `per_page` (*int*) – Number of docs in each *boto.cloudsearch2.search.SearchResults* object.

Return type generator

Returns Generator containing *boto.cloudsearch2.search.SearchResults*

```
get_num_hits (query)
```

Return the total number of hits for query

Parameters `query` (*boto.cloudsearch2.search.Query*) – a group of search criteria

Return type `int`

Returns Total number of hits for query

search (*q=None, parser=None, fq=None, rank=None, return_fields=None, size=10, start=0, facet=None, highlight=None, sort=None, partial=None, options=None*)
Send a query to CloudSearch

Each search query should use at least the `q` or `bq` argument to specify the search parameter. The other options are used to specify the criteria of the search.

Parameters

- **q** (*string*) – A string to search the default search fields for.
- **parser** (*string*) – The parser to use. ‘simple’, ‘structured’, ‘lucene’, ‘dis-max’
- **fq** (*string*) – The filter query to use.
- **sort** (*List of strings*) – A list of fields or rank expressions used to order the search results. Order is handled by adding ‘desc’ or ‘asc’ after the field name. `['year desc', 'author asc']`
- **return_fields** (*List of strings*) – A list of fields which should be returned by the search. If this field is not specified, only IDs will be returned. `['headline']`
- **size** (*int*) – Number of search results to specify
- **start** (*int*) – Offset of the first search result to return (can be used for paging)
- **facet** (*dict*) – Dictionary of fields for which facets should be returned The facet value is string of JSON options `{'year': '{sort:"bucket", size:3}', 'genres': '{buckets:["Action","Adventure","Sci-Fi"]}'}`
- **highlight** (*dict*) – Dictionary of fields for which highlights should be returned The facet value is string of JSON options `{'genres': '{format:'text',max_phrases:2,pre_tag:'', post_tag:''}'}`
- **partial** (*bool*) – Should partial results from a partitioned service be returned if one or more index partitions are unreachable.
- **options** (*str*) – Options for the query parser specified in *parser*. Specified as a string in JSON format. `{fields: ['title^5', 'description']}`

Return type `boto.cloudsearch2.search.SearchResults`

Returns Returns the results of this search

The following examples all assume we have indexed a set of documents with fields: *author, date, headline*

A simple search will look for documents whose default text search fields will contain the search word exactly:

```
>>> search(q='Tim') # Return documents with the word Tim in them (but not
↳ Timothy)
```

A simple search with more keywords will return documents whose default text search fields contain the search strings together or separately.

```
>>> search(q='Tim apple') # Will match "tim" and "apple"
```

More complex searches require the boolean search operator.

Wildcard searches can be used to search for any words that start with the search string.

```
>>> search(q="'Tim*') # Return documents with words like Tim or Timothy
```

Search terms can also be combined. Allowed operators are “and”, “or”, “not”, “field”, “optional”, “token”, “phrase”, or “filter”

```
>>> search(q="(and 'Tim' (field author 'John Smith'))", parser='structured')
```

Facets allow you to show classification information about the search results. For example, you can retrieve the authors who have written about Tim with a max of 3

```
>>> search(q='Tim', facet={'Author': '{sort:"bucket", size:3}'})
```

```
class boto.cloudsearch2.search.SearchResults (**attrs)
```

```
next_page ()
```

Call Cloudsearch to get the next page of search results

Return type *boto.cloudsearch2.search.SearchResults*

Returns the following page of search results

```
exception boto.cloudsearch2.search.SearchServiceException
```

4.114.7 boto.cloudsearch2.document

```
exception boto.cloudsearch2.document.CommitMismatchError
```

```
errors = None
```

```
class boto.cloudsearch2.document.CommitResponse (response, doc_service, sdf,  
                                                signed_request=False)
```

Wrapper for response to Cloudsearch document batch commit.

Parameters

- **response** (*requests.models.Response*) – Response from Cloudsearch /documents/batch API
- **doc_service** (*boto.cloudsearch2.document.DocumentServiceConnection*) – Object containing the documents posted and methods to retry

Raises *boto.exception.BotoServerError*

Raises *boto.cloudsearch2.document.SearchServiceException*

Raises *boto.cloudsearch2.document.EncodingError*

Raises *boto.cloudsearch2.document.ContentTooLongError*

```
exception boto.cloudsearch2.document.ContentTooLongError
```

Content sent for Cloud Search indexing was too long

This will usually happen when documents queued for indexing add up to more than the limit allowed per upload batch (5MB)

```
class boto.cloudsearch2.document.DocumentServiceConnection (domain=None, end-  
                                                         point=None)
```

A CloudSearch document service.

The `DocumentServiceConnection` is used to add, remove and update documents in CloudSearch. Commands are uploaded to CloudSearch in SDF (Search Document Format).

To generate an appropriate SDF, use `add()` to add or update documents, as well as `delete()` to remove documents.

Once the set of documents is ready to be index, use `commit()` to send the commands to CloudSearch.

If there are a lot of documents to index, it may be preferable to split the generation of SDF data and the actual uploading into CloudSearch. Retrieve the current SDF with `get_sdf()`. If this file is the uploaded into S3, it can be retrieved back afterwards for upload into CloudSearch using `add_sdf_from_s3()`.

The SDF is not cleared after a `commit()`. If you wish to continue using the `DocumentServiceConnection` for another batch upload of commands, you will need to `clear_sdf()` first to stop the previous batch of commands from being uploaded again.

add (*_id*, *fields*)

Add a document to be processed by the `DocumentService`

The document will not actually be added until `commit()` is called

Parameters

- **_id** (*string*) – A unique ID used to refer to this document.
- **fields** (*dict*) – A dictionary of key-value pairs to be uploaded .

add_sdf_from_s3 (*key_obj*)

Load an SDF from S3

Using this method will result in documents added through `add()` and `delete()` being ignored.

Parameters **key_obj** (*boto.s3.key.Key*) – An S3 key which contains an SDF

clear_sdf ()

Clear the working documents from this `DocumentServiceConnection`

This should be used after `commit()` if the connection will be reused for another set of documents.

commit ()

Actually send an SDF to CloudSearch for processing

If an SDF file has been explicitly loaded it will be used. Otherwise, documents added through `add()` and `delete()` will be used.

Return type *CommitResponse*

Returns A summary of documents added and deleted

delete (*_id*)

Schedule a document to be removed from the CloudSearch service

The document will not actually be scheduled for removal until `commit()` is called

Parameters **_id** (*string*) – The unique ID of this document.

get_sdf ()

Generate the working set of documents in Search Data Format (SDF)

Return type *string*

Returns JSON-formatted string of the documents in SDF

exception `boto.cloudsearch2.document.EncodingError`

Content sent for Cloud Search indexing was incorrectly encoded.

This usually happens when a document is marked as unicode but non-unicode characters are present.

exception `boto.cloudsearch2.document.SearchServiceException`

4.115 CloudSearch Domain

4.115.1 boto.cloudsearchdomain

`boto.cloudsearchdomain.connect_to_region` (*region_name*, ***kw_params*)

`boto.cloudsearchdomain.regions` ()

Get all available regions for the Amazon CloudSearch Domain service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.115.2 boto.cloudsearchdomain.layer1

class `boto.cloudsearchdomain.layer1.CloudSearchDomainConnection` (***kwargs*)

You use the AmazonCloudSearch2013 API to upload documents to a search domain and search those documents.

The endpoints for submitting *UploadDocuments*, *Search*, and *Suggest* requests are domain-specific. To get the endpoints for your domain, use the Amazon CloudSearch configuration service *DescribeDomains* action. The domain endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console. You submit suggest requests to the search endpoint.

For more information, see the [‘Amazon CloudSearch Developer Guide’](#).

APIVersion = '2013-01-01'

AuthServiceName = 'cloudsearch'

DefaultRegionEndpoint = 'cloudsearch.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

make_request (*verb*, *resource*, *headers=None*, *data=""*, *expected_status=None*, *params=None*)

Makes a request to the server, with stock multiple-retry logic.

search (*query*, *cursor=None*, *expr=None*, *facet=None*, *filter_query=None*, *highlight=None*, *partial=None*, *query_options=None*, *query_parser=None*, *ret=None*, *size=None*, *sort=None*, *start=None*)

Retrieves a list of documents that match the specified search criteria. How you specify the search criteria depends on which query parser you use. Amazon CloudSearch supports four query parsers:

- *simple*: search all *text* and *text-array* fields for the specified string. Search for phrases, individual terms, and prefixes.
- *structured*: search specific fields, construct compound queries using Boolean operators, and use advanced features such as term boosting and proximity searching.
- *lucene*: specify search criteria using the Apache Lucene query parser syntax.
- *dismax*: specify search criteria using the simplified subset of the Apache Lucene query parser syntax defined by the DisMax query parser.

For more information, see [‘Searching Your Data’](#) in the Amazon CloudSearch Developer Guide .

The endpoint for submitting *Search* requests is domain-specific. You submit search requests to a domain’s search endpoint. To get the search endpoint for your domain, use the Amazon CloudSearch configuration service *DescribeDomains* action. A domain’s endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

Parameters `cursor` (*string*) – Retrieves a cursor value you can use to page through large result sets. Use the `size` parameter to control the number of hits to include in each response. You can specify either the `cursor` or `start` parameter in a request; they are mutually exclusive. To get the first cursor, set the cursor value to `initial`. In subsequent requests, specify the cursor value returned in the hits section of the response.

For more information, see [‘Paginating Results’_ in the Amazon CloudSearch Developer Guide](#) .

Parameters `expr` (*string*) – Defines one or more numeric expressions that can be used to sort results or specify search or filter criteria. You can also specify expressions as return fields.

For more information about defining and using expressions, see [‘Configuring Expressions’_ in the Amazon CloudSearch Developer Guide](#) .

Parameters `facet` (*string*) – Specifies one or more fields for which to get facet information, and options that control how the facet information is returned. Each specified field must be facet-enabled in the domain configuration. The fields and options are specified in JSON using the form `{“FIELD”:{“OPTION”:VALUE,“OPTION”:"STRING"},“FIELD”:{“OPTION”:VALUE,“OPTION”:"STRING"}}`.

You can specify the following faceting options:

- **`buckets` specifies an array of the facet values or ranges to count.** Ranges are specified using the same syntax that you use to search for a range of values. For more information, see [‘Searching for a Range of Values’_ in the Amazon CloudSearch Developer Guide](#) . Buckets are returned in the order they are specified in the request. The `sort` and `size` options are not valid if you specify `buckets`.
- **`size` specifies the maximum number of facets to include in the results.** By default, Amazon CloudSearch returns counts for the top 10. The `size` parameter is only valid when you specify the `sort` option; it cannot be used in conjunction with `buckets`.
- **`sort` specifies how you want to sort the facets in the results:** `bucket` or `count`. Specify `bucket` to sort alphabetically or numerically by facet value (in ascending order). Specify `count` to sort by the facet counts computed for each facet value (in descending order). To retrieve facet counts for particular values or ranges of values, use the `buckets` option instead of `sort`.

If no facet options are specified, facet counts are computed for all field values, the facets are sorted by facet count, and the top 10 facets are returned in the results.

For more information, see [‘Getting and Using Facet Information’_ in the Amazon CloudSearch Developer Guide](#) .

Parameters `filter_query` (*string*) – Specifies a structured query that filters the results of a search without affecting how the results are scored and sorted. You use `filterQuery` in conjunction with the `query` parameter to filter the documents that match the constraints specified in the `query` parameter. Specifying a filter controls only which matching documents are included in the results, it has no effect on how they are scored and sorted. The `filterQuery` parameter supports the full structured query syntax.

For more information about using filters, see [‘Filtering Matching Documents’_ in the Amazon CloudSearch Developer Guide](#) .

Parameters `highlight` (*string*) – Retrieves highlights for matches in the specified `text` or `text-array` fields. Each specified field must be highlight enabled in the domain configuration. The fields and options are specified in JSON using the form `{“FIELD”:{“OPTION”:VALUE,“OPTION”:"STRING"},“FIELD”:{“OPTION”:VALUE,“OPTION”:"STRING"}}`.

You can specify the following highlight options:

- **format:** specifies the format of the data in the text field: *text* or *html*. When data is returned as HTML, all non-alphanumeric characters are encoded. The default is *html*.
- **max_phrases:** specifies the maximum number of occurrences of the search term(s) you want to highlight. By default, the first occurrence is highlighted.
- **pre_tag:** specifies the string to prepend to an occurrence of a search term. The default for HTML highlights is ``. The default for text highlights is `*`.
- **post_tag:** specifies the string to append to an occurrence of a search term. The default for HTML highlights is ``. The default for text highlights is `*`.

If no highlight options are specified for a field, the returned field text is treated as HTML and the first match is highlighted with emphasis tags: `search-term`.

Parameters

- **partial** (*boolean*) – Enables partial results to be returned if one or more index partitions are unavailable. When your search index is partitioned across multiple search instances, by default Amazon CloudSearch only returns results if every partition can be queried. This means that the failure of a single search instance can result in 5xx (internal server) errors. When you enable partial results, Amazon CloudSearch returns whatever results are available and includes the percentage of documents searched in the search results (percent-searched). This enables you to more gracefully degrade your users' search experience. For example, rather than displaying no results, you could display the partial results and a message indicating that the results might be incomplete due to a temporary system outage.
- **query** (*string*) – Specifies the search criteria for the request. How you specify the search criteria depends on the query parser used for the request and the parser options specified in the *queryOptions* parameter. By default, the *simple* query parser is used to process requests. To use the *structured*, *lucene*, or *dismax* query parser, you must also specify the *queryParser* parameter.

For more information about specifying search criteria, see ‘Searching Your Data’ in the Amazon CloudSearch Developer Guide .

Parameters *query_options* (*string*) –

Configures options for the query parser specified in the *queryParser* parameter.

The options you can configure vary according to which parser you use:

- **defaultOperator:** The default operator used to combine individual terms in the search string. For example: *defaultOperator*: ‘or’. For the *dismax* parser, you specify a percentage that represents the percentage of terms in the search string (rounded down) that must match, rather than a default operator. A value of 0% is the equivalent to OR, and a value of 100% is equivalent to AND. The percentage must be specified as a value in the range 0-100 followed by the percent (%) symbol. For example, *defaultOperator*: 50%. Valid values: *and*, *or*, a percentage in the range 0%-100% (*dismax*). Default: *and* (*simple*, *structured*, *lucene*) or *100* (*dismax*). Valid for: *simple*, *structured*, *lucene*, and *dismax*.
- **fields:** An array of the fields to search when no fields are specified in a search. If no fields are specified in a search and this option is not specified, all text and text-array fields are searched. You can specify a weight for each field to control the relative importance of each field when Amazon CloudSearch calculates relevance scores. To specify a field weight, append a caret (^) symbol and the weight to the field name. For example, to boost the importance of the *title* field over the *description* field you could specify: `“fields”:[“title^5”,“description”]`. Valid values: The name of any configured field and an optional numeric value greater than zero. Default: All *text* and *text- array* fields. Valid for: *simple*, *structured*, *lucene*, and *dismax*.
- **operators:** An array of the operators or special characters you want to disable for the simple query parser. If you disable the *and*, *or*, or *not* operators, the corresponding operators (

+, |, -) have no special meaning and are dropped from the search string. Similarly, disabling *prefix* disables the wildcard operator (***) and disabling *phrase* disables the ability to search for phrases by enclosing phrases in double quotes. Disabling *precedence* disables the ability to control order of precedence using parentheses. Disabling *near* disables the ability to use the *~* operator to perform a sloppy phrase search. Disabling the *fuzzy* operator disables the ability to use the *~* operator to perform a fuzzy search. *escape* disables the ability to use a backslash (*'*) to *escape special characters within the search string*. *Disabling whitespace is an advanced option that prevents the parser from tokenizing on whitespace, which can be useful for Vietnamese. (It prevents Vietnamese words from being split incorrectly.) For example, you could disable all operators other than the phrase operator to support just simple term and phrase queries: "operators":["and","not","or","prefix"]*. Valid values: *and, escape, fuzzy, near, not, or, phrase, precedence, prefix, whitespace*. Default: All operators and special characters are enabled. Valid for: *simple*.

- ***phraseFields***: An array of the *text* or *text-array* fields you want to use for phrase searches. When the terms in the search string appear in close proximity within a field, the field scores higher. You can specify a weight for each field to boost that score. The *phraseSlop* option controls how much the matches can deviate from the search string and still be boosted. To specify a field weight, append a caret (*^*) symbol and the weight to the field name. For example, to boost phrase matches in the *title* field over the *abstract* field, you could specify: *"phraseFields":["title^3","plot"]* Valid values: The name of any *text* or *text-array* field and an optional numeric value greater than zero. Default: No fields. If you don't specify any fields with *phraseFields*, proximity scoring is disabled even if *phraseSlop* is specified. Valid for: *dismax*.
- ***phraseSlop***: An integer value that specifies how much matches can deviate from the search phrase and still be boosted according to the weights specified in the *phraseFields* option; for example, *phraseSlop: 2*. You must also specify *phraseFields* to enable proximity scoring. Valid values: positive integers. Default: 0. Valid for: *dismax*.
- ***explicitPhraseSlop***: An integer value that specifies how much a match can deviate from the search phrase when the phrase is enclosed in double quotes in the search string. (Phrases that exceed this proximity distance are not considered a match.) For example, to specify a slop of three for *dismax* phrase queries, you would specify *"explicitPhraseSlop":3*. Valid values: positive integers. Default: 0. Valid for: *dismax*.
- ***tieBreaker***: When a term in the search string is found in a document's field, a score is calculated for that field based on how common the word is in that field compared to other documents. If the term occurs in multiple fields within a document, by default only the highest scoring field contributes to the document's overall score. You can specify a *tieBreaker* value to enable the matches in lower-scoring fields to contribute to the document's score. That way, if two documents have the same max field score for a particular term, the score for the document that has matches in more fields will be higher. The formula for calculating the score with a *tieBreaker* is $(max\ field\ score) + (tieBreaker) * (sum\ of\ the\ scores\ for\ the\ rest\ of\ the\ matching\ fields)$. Set *tieBreaker* to 0 to disregard all but the highest scoring field (pure max): *"tieBreaker":0*. Set to 1 to sum the scores from all fields (pure sum): *"tieBreaker":1*. Valid values: 0.0 to 1.0. Default: 0.0. Valid for: *dismax*.

Parameters ***query_parser*** (*string*)–

Specifies which query parser to use to process the request. If *queryParser* is not specified, Amazon CloudSearch uses the *simple* query parser.

Amazon CloudSearch supports four query parsers:

- ***simple***: perform simple searches of *text* and *text-array* fields. By default, the *simple* query parser searches all *text* and *text-array* fields. You can specify which fields to search by with the *queryOptions* parameter. If you prefix a search term with a plus sign (+) documents must contain the term to be considered a match. (This is the default, unless you configure the default operator with the *queryOptions* parameter.) You can use the - (NOT), | (OR), and *

(wildcard) operators to exclude particular terms, find results that match any of the specified terms, or search for a prefix. To search for a phrase rather than individual terms, enclose the phrase in double quotes. For more information, see [‘Searching for Text’](#) in the Amazon CloudSearch Developer Guide .

- **structured**: perform advanced searches by combining multiple expressions to define the search criteria. You can also search within particular fields, search for values and ranges of values, and use advanced options such as term boosting, *matchall*, and *near*. For more information, see [‘Constructing Compound Queries’](#) in the Amazon CloudSearch Developer Guide .
- **lucene**: search using the Apache Lucene query parser syntax. For more information, see [‘Apache Lucene Query Parser Syntax’](#) .
- **dismax**: search using the simplified subset of the Apache Lucene query parser syntax defined by the DisMax query parser. For more information, see [‘DisMax Query Parser Syntax’](#) .

Parameters

- **ret** (*string*) – Specifies the field and expression values to include in the response. Multiple fields or expressions are specified as a comma-separated list. By default, a search response includes all return enabled fields (*_all_fields*). To return only the document IDs for the matching documents, specify *_no_fields* . To retrieve the relevance score calculated for each document, specify *_score* .
- **size** (*long*) – Specifies the maximum number of search hits to include in the response.
- **sort** (*string*) – Specifies the fields or custom expressions to use to sort the search results. Multiple fields or expressions are specified as a comma-separated list. You must specify the sort direction (*asc* or *desc*) for each field; for example, *year desc,title asc* . To use a field to sort results, the field must be sort-enabled in the domain configuration. Array type fields cannot be used for sorting. If no *sort* parameter is specified, results are sorted by their default relevance scores in descending order: *_score desc* . You can also sort by document ID (*_id asc*) and version (*_version desc*).

For more information, see [‘Sorting Results’](#) in the Amazon CloudSearch Developer Guide .

Parameters start (*long*) – Specifies the offset of the first search hit you want to return. Note that the result set is zero-based; the first result is at index 0. You can specify either the *start* or *cursor* parameter in a request, they are mutually exclusive.

For more information, see [‘Paginating Results’](#) in the Amazon CloudSearch Developer Guide .

suggest (*query, suggester, size=None*)

Retrieves autocomplete suggestions for a partial query string. You can use suggestions enable you to display likely matches before users finish typing. In Amazon CloudSearch, suggestions are based on the contents of a particular text field. When you request suggestions, Amazon CloudSearch finds all of the documents whose values in the suggester field start with the specified query string. The beginning of the field must match the query string to be considered a match.

For more information about configuring suggesters and retrieving suggestions, see [‘Getting Suggestions’](#) in the Amazon CloudSearch Developer Guide .

The endpoint for submitting *Suggest* requests is domain- specific. You submit suggest requests to a domain’s search endpoint. To get the search endpoint for your domain, use the Amazon CloudSearch configuration service *DescribeDomains* action. A domain’s endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

Parameters

- **query** (*string*) – Specifies the string for which you want to get suggestions.

- **suggester** (*string*) – Specifies the name of the suggester to use to find suggested matches.
- **size** (*long*) – Specifies the maximum number of suggestions to return.

upload_documents (*documents, content_type*)

Posts a batch of documents to a search domain for indexing. A document batch is a collection of add and delete operations that represent the documents you want to add, update, or delete from your domain. Batches can be described in either JSON or XML. Each item that you want Amazon CloudSearch to return as a search result (such as a product) is represented as a document. Every document has a unique ID and one or more fields that contain the data that you want to search and return in results. Individual documents cannot contain more than 1 MB of data. The entire batch cannot exceed 5 MB. To get the best possible upload performance, group add and delete operations in batches that are close the 5 MB limit. Submitting a large volume of single-document batches can overload a domain's document service.

The endpoint for submitting *UploadDocuments* requests is domain-specific. To get the document endpoint for your domain, use the Amazon CloudSearch configuration service *DescribeDomains* action. A domain's endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console.

For more information about formatting your data for Amazon CloudSearch, see '[Preparing Your Data](#)' in the Amazon CloudSearch Developer Guide . For more information about uploading data for indexing, see '[Uploading Data](#)' in the Amazon CloudSearch Developer Guide .

Parameters

- **documents** (*blob*) – A batch of documents formatted in JSON or HTML.
- **content_type** (*string*) –

The format of the batch you are uploading. Amazon CloudSearch supports two document batch formats:

- application/json
- application/xml

4.115.3 boto.cloudsearchdomain.exceptions

```
exception boto.cloudsearchdomain.exceptions.DocumentServiceException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.cloudsearchdomain.exceptions.SearchException (status,      reason,
                                                                    body=None, *args)
```

4.116 CloudTrail

4.116.1 boto.cloudtrail

```
boto.cloudtrail.connect_to_region (region_name, **kw_params)
```

```
boto.cloudtrail.regions ()
```

Get all available regions for the AWS Cloudtrail service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.116.2 boto.cloudtrail.layer1

class `boto.cloudtrail.layer1.CloudTrailConnection` (**kwargs)

AWS CloudTrail This is the CloudTrail API Reference. It provides descriptions of actions, data types, common parameters, and common errors for CloudTrail.

CloudTrail is a web service that records AWS API calls for your AWS account and delivers log files to an Amazon S3 bucket. The recorded information includes the identity of the user, the start time of the AWS API call, the source IP address, the request parameters, and the response elements returned by the service.

As an alternative to using the API, you can use one of the AWS SDKs, which consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWS CloudTrail. For example, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For information about the AWS SDKs, including how to download and install them, see the [‘Tools for Amazon Web Services page’](#).

See the CloudTrail User Guide for information about the data that is included with each AWS API call listed in the log files.

APIVersion = '2013-11-01'

DefaultRegionEndpoint = 'cloudtrail.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

ServiceName = 'CloudTrail'

TargetPrefix = 'com.amazonaws.cloudtrail.v20131101.CloudTrail_20131101'

create_trail (*name*, *s3_bucket_name*, *s3_key_prefix=None*, *sns_topic_name=None*, *include_global_service_events=None*, *cloud_watch_logs_log_group_arn=None*, *cloud_watch_logs_role_arn=None*)

From the command line, use `create-subscription`.

Creates a trail that specifies the settings for delivery of log data to an Amazon S3 bucket.

Parameters

- **name** (*string*) – Specifies the name of the trail.
- **s3_bucket_name** (*string*) – Specifies the name of the Amazon S3 bucket designated for publishing log files.
- **s3_key_prefix** (*string*) – Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.
- **sns_topic_name** (*string*) – Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **include_global_service_events** (*boolean*) – Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **cloud_watch_logs_log_group_arn** (*string*) – Specifies a log group name using an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered. Not required unless you specify `CloudWatchLogsRoleArn`.
- **cloud_watch_logs_role_arn** (*string*) – Specifies the role for the CloudWatch Logs endpoint to assume to write to a users log group.

delete_trail (*name*)

Deletes a trail.

Parameters **name** (*string*) – The name of a trail to be deleted.

describe_trails (*trail_name_list=None*)

Retrieves settings for the trail associated with the current region for your account.

Parameters **trail_name_list** (*list*) – The trail returned.

get_trail_status (*name*)

Returns a JSON-formatted list of information about the specified trail. Fields include information on delivery errors, Amazon SNS and Amazon S3 errors, and start and stop logging times for each trail.

Parameters **name** (*string*) – The name of the trail for which you are requesting the current status.

lookup_events (*lookup_attributes=None, start_time=None, end_time=None, max_results=None, next_token=None*)

Looks up API activity events captured by CloudTrail that create, update, or delete resources in your account. Events for a region can be looked up for the times in which you had CloudTrail turned on in that region during the last seven days. Lookup supports five different attributes: time range (defined by a start time and end time), user name, event name, resource type, and resource name. All attributes are optional. The maximum number of attributes that can be specified in any one lookup request are time range and one other attribute. The default number of results returned is 10, with a maximum of 50 possible. The response includes a token that you can use to get the next page of results. The rate of lookup requests is limited to one per second per account. If this limit is exceeded, a throttling error occurs. Events that occurred during the selected time range will not be available for lookup if CloudTrail logging was not enabled when the events occurred.

Parameters

- **lookup_attributes** (*list*) – Contains a list of lookup attributes. Currently the list can contain only one item.
- **start_time** (*timestamp*) – Specifies that only events that occur after or at the specified time are returned. If the specified start time is after the specified end time, an error is returned.
- **end_time** (*timestamp*) – Specifies that only events that occur before or at the specified time are returned. If the specified end time is before the specified start time, an error is returned.
- **max_results** (*integer*) – The number of events to return. Possible values are 1 through 50. The default is 10.
- **next_token** (*string*) – The token to use to get the next page of results after a previous API call. This token must be passed in with the same parameters that were specified in the the original call. For example, if the original call specified an AttributeKey of 'Username' with a value of 'root', the call with NextToken should include those same parameters.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

start_logging (*name*)

Starts the recording of AWS API calls and log file delivery for a trail.

Parameters **name** (*string*) – The name of the trail for which CloudTrail logs AWS API calls.

stop_logging (*name*)

Suspends the recording of AWS API calls and log file delivery for the specified trail. Under most circumstances, there is no need to use this action. You can update a trail without stopping it first. This action is the only way to stop recording.

Parameters **name** (*string*) – Communicates to CloudTrail the name of the trail for which to stop logging AWS API calls.

update_trail (*name*, *s3_bucket_name=None*, *s3_key_prefix=None*, *sns_topic_name=None*, *include_global_service_events=None*, *cloud_watch_logs_log_group_arn=None*, *cloud_watch_logs_role_arn=None*)

From the command line, use *update-subscription*.

Updates the settings that specify delivery of log files. Changes to a trail do not require stopping the CloudTrail service. Use this action to designate an existing bucket for log delivery. If the existing bucket has previously been a target for CloudTrail log files, an IAM policy exists for the bucket.

Parameters

- **name** (*string*) – Specifies the name of the trail.
- **s3_bucket_name** (*string*) – Specifies the name of the Amazon S3 bucket designated for publishing log files.
- **s3_key_prefix** (*string*) – Specifies the Amazon S3 key prefix that precedes the name of the bucket you have designated for log file delivery.
- **sns_topic_name** (*string*) – Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- **include_global_service_events** (*boolean*) – Specifies whether the trail is publishing events from global services such as IAM to the log files.
- **cloud_watch_logs_log_group_arn** (*string*) – Specifies a log group name using an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered. Not required unless you specify `CloudWatchLogsRoleArn`.
- **cloud_watch_logs_role_arn** (*string*) – Specifies the role for the CloudWatch Logs endpoint to assume to write to a users log group.

4.116.3 boto.cloudtrail.exceptions

Exceptions that are specific to the cloudtrail module.

exception `boto.cloudtrail.exceptions.CloudWatchLogsDeliveryUnavailableException` (*status*, *reason*, *body=None*, **args*)

exception `boto.cloudtrail.exceptions.InsufficientS3BucketPolicyException` (*status*, *reason*, *body=None*, **args*)

Raised when the S3 bucket does not allow Cloudtrail to write files into the prefix.

exception `boto.cloudtrail.exceptions.InsufficientSnsTopicPolicyException` (*status*, *reason*, *body=None*, **args*)

Raised when the SNS topic does not allow Cloudtrail to post messages.

exception `boto.cloudtrail.exceptions.InternalErrorException` (*status*, *reason*,
body=None, *args)

Raised when there was an internal Cloudtrail error.

exception `boto.cloudtrail.exceptions.InvalidCloudWatchLogsLogGroupArnException` (*status*,
reason,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidCloudWatchLogsRoleArnException` (*status*,
reason,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidLookupAttributesException` (*status*,
reason,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidMaxResultsException` (*status*, *reason*,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidNextTokenException` (*status*, *reason*,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidS3BucketNameException` (*status*,
reason,
body=None,
*args)

Raised when an invalid S3 bucket name is passed to Cloudtrail.

exception `boto.cloudtrail.exceptions.InvalidS3PrefixException` (*status*, *reason*,
body=None,
*args)

Raised when an invalid key prefix is given.

exception `boto.cloudtrail.exceptions.InvalidSnsTopicNameException` (*status*,
reason,
body=None,
*args)

Raised when an invalid SNS topic name is passed to Cloudtrail.

exception `boto.cloudtrail.exceptions.InvalidTimeRangeException` (*status*, *reason*,
body=None,
*args)

exception `boto.cloudtrail.exceptions.InvalidTrailNameException` (*status*, *reason*,
body=None,
*args)

Raised when the trail name is invalid.

exception `boto.cloudtrail.exceptions.MaximumNumberOfTrailsExceededException` (*status*,
reason,
body=None,
*args)

Raised when no more trails can be created.

```
exception boto.cloudtrail.exceptions.S3BucketDoesNotExistException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

Raised when the given S3 bucket does not exist.

```
exception boto.cloudtrail.exceptions.TrailAlreadyExistsException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

Raised when the given trail name already exists.

```
exception boto.cloudtrail.exceptions.TrailNotFoundException (status,      reason,  
                                                                    body=None, *args)
```

Raised when the given trail name is not found.

```
exception boto.cloudtrail.exceptions.TrailNotProvidedException (status,  reason,  
                                                                    body=None,  
                                                                    *args)
```

Raised when no trail name was provided.

4.117 CloudWatch Reference

4.117.1 boto.ec2.cloudwatch

This module provides an interface to the Elastic Compute Cloud (EC2) CloudWatch service from AWS.

```
class boto.ec2.cloudwatch.CloudWatchConnection (aws_access_key_id=None,  
                                                aws_secret_access_key=None,  
                                                is_secure=True,           port=None,  
                                                proxy=None,           proxy_port=None,  
                                                proxy_user=None,  
                                                proxy_pass=None,       debug=0,  
                                                https_connection_factory=None,  
                                                region=None,   path='/',   security_token=None, validate_certs=True,  
                                                profile_name=None)
```

Init method to create a new connection to EC2 Monitoring Service.

B{Note:} The host argument is overridden by the host specified in the boto configuration file.

```
APIVersion = '2010-08-01'
```

```
DefaultRegionEndpoint = 'monitoring.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
build_dimension_param (dimension, params)
```

```
build_list_params (params, items, label)
```

```
build_put_params (params, name, value=None, timestamp=None, unit=None, dimensions=None,  
                  statistics=None)
```

```
create_alarm (alarm)
```

Creates or updates an alarm and associates it with the specified Amazon CloudWatch metric. Optionally, this operation can associate one or more Amazon Simple Notification Service resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is evaluated and its `StateValue` is set appropriately. Any actions associated with the `StateValue` is then executed.

When updating an existing alarm, its `StateValue` is left unchanged.

Parameters `alarm` (`boto.ec2.cloudwatch.alarm.MetricAlarm`) – `MetricAlarm` object.

delete_alarms (`alarms`)

Deletes all specified alarms. In the event of an error, no alarms are deleted.

Parameters `alarms` (`list`) – List of alarm names.

describe_alarm_history (`alarm_name=None`, `start_date=None`, `end_date=None`, `max_records=None`, `history_item_type=None`, `next_token=None`)

Retrieves history for the specified alarm. Filter alarms by date range or item type. If an alarm name is not specified, Amazon CloudWatch returns histories for all of the owner's alarms.

Amazon CloudWatch retains the history of deleted alarms for a period of six weeks. If an alarm has been deleted, its history can still be queried.

Parameters

- **alarm_name** (`string`) – The name of the alarm.
- **start_date** (`datetime`) – The starting date to retrieve alarm history.
- **end_date** (`datetime`) – The starting date to retrieve alarm history.
- **history_item_type** (`string`) – The type of alarm histories to retrieve (`ConfigurationUpdate` | `StateUpdate` | `Action`)
- **max_records** (`int`) – The maximum number of alarm descriptions to retrieve.
- **next_token** (`string`) – The token returned by a previous call to indicate that there is more data.

:rtype list

describe_alarms (`action_prefix=None`, `alarm_name_prefix=None`, `alarm_names=None`, `max_records=None`, `state_value=None`, `next_token=None`)

Retrieves alarms with the specified names. If no name is specified, all alarms for the user are returned. Alarms can be retrieved by using only a prefix for the alarm name, the alarm state, or a prefix for any action.

Parameters

- **action_prefix** (`string`) – The action name prefix.
- **alarm_name_prefix** (`string`) – The alarm name prefix. `AlarmNames` cannot be specified if this parameter is specified.
- **alarm_names** (`list`) – A list of alarm names to retrieve information for.
- **max_records** (`int`) – The maximum number of alarm descriptions to retrieve.
- **state_value** (`string`) – The state value to be used in matching alarms.
- **next_token** (`string`) – The token returned by a previous call to indicate that there is more data.

:rtype list

describe_alarms_for_metric (`metric_name`, `namespace`, `period=None`, `statistic=None`, `dimensions=None`, `unit=None`)

Retrieves all alarms for a single metric. Specify a statistic, period, or unit to filter the set of alarms further.

Parameters

- **metric_name** (`string`) – The name of the metric.

- **namespace** (*string*) – The namespace of the metric.
- **period** (*int*) – The period in seconds over which the statistic is applied.
- **statistic** (*string*) – The statistic for the metric.
- **dimensions** (*dict*) – A dictionary containing name/value pairs that will be used to filter the results. The key in the dictionary is the name of a Dimension. The value in the dictionary is either a scalar value of that Dimension name that you want to filter on, a list of values to filter on or None if you want all metrics with that Dimension name.

:rtype list

disable_alarm_actions (*alarm_names*)

Disables actions for the specified alarms.

Parameters **alarms** (*list*) – List of alarm names.

enable_alarm_actions (*alarm_names*)

Enables actions for the specified alarms.

Parameters **alarms** (*list*) – List of alarm names.

get_metric_statistics (*period, start_time, end_time, metric_name, namespace, statistics, dimensions=None, unit=None*)

Get time-series data for one or more statistics of a given metric.

Parameters

- **period** (*integer*) – The granularity, in seconds, of the returned datapoints. Period must be at least 60 seconds and must be a multiple of 60. The default value is 60.
- **start_time** (*datetime*) – The time stamp to use for determining the first datapoint to return. The value specified is inclusive; results include datapoints with the time stamp specified.
- **end_time** (*datetime*) – The time stamp to use for determining the last datapoint to return. The value specified is exclusive; results will include datapoints up to the time stamp specified.
- **metric_name** (*string*) – The metric name.
- **namespace** (*string*) – The metric’s namespace.
- **statistics** (*list*) – A list of statistics names Valid values: Average | Sum | SampleCount | Maximum | Minimum
- **dimensions** (*dict*) – A dictionary of dimension key/values where the key is the dimension name and the value is either a scalar value or an iterator of values to be associated with that dimension.
- **unit** (*string*) – The unit for the metric. Value values are: Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None

Return type *list*

list_metrics (*next_token=None, dimensions=None, metric_name=None, namespace=None*)

Returns a list of the valid metrics for which there is recorded data available.

Parameters

- **next_token** (*str*) – A maximum of 500 metrics will be returned at one time. If more results are available, the ResultSet returned will contain a non-Null next_token attribute. Passing that token as a parameter to list_metrics will retrieve the next page of metrics.
- **dimensions** (*dict*) – A dictionary containing name/value pairs that will be used to filter the results. The key in the dictionary is the name of a Dimension. The value in the dictionary is either a scalar value of that Dimension name that you want to filter on or None if you want all metrics with that Dimension name. To be included in the result a metric must contain all specified dimensions, although the metric may contain additional dimensions beyond the requested metrics. The Dimension names, and values must be strings between 1 and 250 characters long. A maximum of 10 dimensions are allowed.
- **metric_name** (*str*) – The name of the Metric to filter against. If None, all Metric names will be returned.
- **namespace** (*str*) – A Metric namespace to filter against (e.g. AWS/EC2). If None, Metrics from all namespaces will be returned.

put_metric_alarm (*alarm*)

Creates or updates an alarm and associates it with the specified Amazon CloudWatch metric. Optionally, this operation can associate one or more Amazon Simple Notification Service resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to INSUFFICIENT_DATA. The alarm is evaluated and its StateValue is set appropriately. Any actions associated with the StateValue is then executed.

When updating an existing alarm, its StateValue is left unchanged.

Parameters **alarm** (`boto.ec2.cloudwatch.alarm.MetricAlarm`) – MetricAlarm object.

put_metric_data (*namespace, name, value=None, timestamp=None, unit=None, dimensions=None, statistics=None*)

Publishes metric data points to Amazon CloudWatch. Amazon Cloudwatch associates the data points with the specified metric. If the specified metric does not exist, Amazon CloudWatch creates the metric. If a list is specified for some, but not all, of the arguments, the remaining arguments are repeated a corresponding number of times.

Parameters

- **namespace** (*str*) – The namespace of the metric.
- **name** (*str or list*) – The name of the metric.
- **value** (*float or list*) – The value for the metric.
- **timestamp** (*datetime or list*) – The time stamp used for the metric. If not specified, the default value is set to the time the metric data was received.
- **unit** (*string or list*) – The unit of the metric. Valid Values: Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None
- **dimensions** (*dict*) – Add extra name value pairs to associate with the metric, i.e.: {'name1': value1, 'name2': (value2, value3)}
- **statistics** (*dict or list*) – Use a statistic set instead of a value, for example:

```
{'maximum': 30, 'minimum': 1, 'samplecount': 100, 'sum': ↵
↵10000}
```

set_alarm_state (*alarm_name*, *state_reason*, *state_value*, *state_reason_data*=None)

Temporarily sets the state of an alarm. When the updated StateValue differs from the previous value, the action configured for the appropriate state is invoked. This is not a permanent change. The next periodic alarm check (in about a minute) will set the alarm to its actual state.

Parameters

- **alarm_name** (*string*) – Descriptive name for alarm.
- **state_reason** (*string*) – Human readable reason.
- **state_value** (*string*) – OK | ALARM | INSUFFICIENT_DATA
- **state_reason_data** (*string*) – Reason string (will be jsonified).

update_alarm (*alarm*)

Creates or updates an alarm and associates it with the specified Amazon CloudWatch metric. Optionally, this operation can associate one or more Amazon Simple Notification Service resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to INSUFFICIENT_DATA. The alarm is evaluated and its StateValue is set appropriately. Any actions associated with the StateValue is then executed.

When updating an existing alarm, its StateValue is left unchanged.

Parameters **alarm** (`boto.ec2.cloudwatch.alarm.MetricAlarm`) – MetricAlarm object.

`boto.ec2.cloudwatch.connect_to_region` (*region_name*, ****kw_params**)

Given a valid region name, return a `boto.ec2.cloudwatch.CloudWatchConnection`.

Parameters **region_name** (*str*) – The name of the region to connect to.

Return type `boto.ec2.CloudWatchConnection` or None

Returns A connection to the given region, or None if an invalid region name is given

`boto.ec2.cloudwatch.regions` ()

Get all available regions for the CloudWatch service.

Return type `list`

Returns A list of `boto.RegionInfo` instances

4.117.2 boto.ec2.cloudwatch.datapoint

class `boto.ec2.cloudwatch.datapoint.Datapoint` (*connection*=None)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

4.117.3 boto.ec2.cloudwatch.metric

class `boto.ec2.cloudwatch.metric.Metric` (*connection*=None)

Statistics = ['Minimum', 'Maximum', 'Sum', 'Average', 'SampleCount']

Units = ['Seconds', 'Microseconds', 'Milliseconds', 'Bytes', 'Kilobytes', 'Megabytes',

create_alarm (*name, comparison, threshold, period, evaluation_periods, statistic, enabled=True, description=None, dimensions=None, alarm_actions=None, ok_actions=None, insufficient_data_actions=None, unit=None*)

Creates or updates an alarm and associates it with this metric. Optionally, this operation can associate one or more Amazon Simple Notification Service resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is evaluated and its `StateValue` is set appropriately. Any actions associated with the `StateValue` is then executed.

When updating an existing alarm, its `StateValue` is left unchanged.

Parameters alarm (`boto.ec2.cloudwatch.alarm.MetricAlarm`) – `MetricAlarm` object.

describe_alarms (*period=None, statistic=None, dimensions=None, unit=None*)

Retrieves all alarms for this metric. Specify a statistic, period, or unit to filter the set of alarms further.

Parameters

- **period** (*int*) – The period in seconds over which the statistic is applied.
- **statistic** (*string*) – The statistic for the metric.
- **dimension** – A dictionary containing name/value pairs that will be used to filter the results. The key in the dictionary is the name of a Dimension. The value in the dictionary is either a scalar value of that Dimension name that you want to filter on, a list of values to filter on or `None` if you want all metrics with that Dimension name.

:rtype list

endElement (*name, value, connection*)

query (*start_time, end_time, statistics, unit=None, period=60*)

Parameters

- **start_time** (*datetime*) – The time stamp to use for determining the first datapoint to return. The value specified is inclusive; results include datapoints with the time stamp specified.
- **end_time** (*datetime*) – The time stamp to use for determining the last datapoint to return. The value specified is exclusive; results will include datapoints up to the time stamp specified.
- **statistics** (*list*) – A list of statistics names Valid values: Average | Sum | SampleCount | Maximum | Minimum
- **unit** (*string*) – The unit for the metric. Value values are: Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None
- **period** (*integer*) – The granularity, in seconds, of the returned datapoints. Period must be at least 60 seconds and must be a multiple of 60. The default value is 60.

startElement (*name, attrs, connection*)

4.117.4 boto.ec2.cloudwatch.alarm

```
class boto.ec2.cloudwatch.alarm.AlarmHistoryItem (connection=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.ec2.cloudwatch.alarm.MetricAlarm (connection=None, name=None, met-  

ric=None, namespace=None, statis-  

tic=None, comparison=None, thresh-  

old=None, period=None, evalu-  

ation_periods=None, unit=None,  

description="", dimensions=None,  

alarm_actions=None, insuf-  

ficient_data_actions=None,  

ok_actions=None)
```

Creates a new Alarm.

Parameters

- **name** (*str*) – Name of alarm.
- **metric** (*str*) – Name of alarm’s associated metric.
- **namespace** (*str*) – The namespace for the alarm’s metric.
- **statistic** (*str*) – The statistic to apply to the alarm’s associated metric. Valid values: SampleCount|Average|Sum|Minimum|Maximum
- **comparison** (*str*) – Comparison used to compare statistic with threshold. Valid values: >=|>|<|<=
- **threshold** (*float*) – The value against which the specified statistic is compared.
- **period** (*int*) – The period in seconds over which the specified statistic is applied.
- **evaluation_periods** (*int*) – The number of periods over which data is compared to the specified threshold.
- **unit** (*str*) – Allowed Values are: Seconds|Microseconds|Milliseconds, Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes, Bits|Kilobits|Megabits|Gigabits|Terabits, Percent|Count| Bytes/Second|Kilobytes/Second|Megabytes/Second| Giga-bytes/Second|Terabytes/Second, Bits/Second|Kilobits/Second|Megabits/Second, Gigabits/Second|Terabits/Second|Count/Second|None
- **description** (*str*) – Description of MetricAlarm
- **dimensions** (*dict*) – A dictionary of dimension key/values where the key is the dimension name and the value is either a scalar value or an iterator of values to be associated with that dimension. Example: {


```
        'InstanceId': ['i-0123456', 'i-0123457'], 'LoadBalancerName': 'test-lb'
      }
```
- **alarm_actions** (*list of strs*) – A list of the ARNs of the actions to take in ALARM state
- **insufficient_data_actions** (*list of strs*) – A list of the ARNs of the actions to take in INSUFFICIENT_DATA state
- **ok_actions** (*list of strs*) – A list of the ARNs of the actions to take in OK state

ALARM = 'ALARM'

INSUFFICIENT_DATA = 'INSUFFICIENT_DATA'

OK = 'OK'

add_alarm_action (*action_arn=None*)

Adds an alarm action, represented as an SNS topic, to this alarm. What do do when alarm is triggered.

Parameters **action_arn** (*str*) – SNS topics to which notification should be sent if the alarm goes to state ALARM.

add_insufficient_data_action (*action_arn=None*)

Adds an insufficient_data action, represented as an SNS topic, to this alarm. What to do when the insufficient_data state is reached.

Parameters **action_arn** (*str*) – SNS topics to which notification should be sent if the alarm goes to state INSUFFICIENT_DATA.

add_ok_action (*action_arn=None*)

Adds an ok action, represented as an SNS topic, to this alarm. What to do when the ok state is reached.

Parameters **action_arn** (*str*) – SNS topics to which notification should be sent if the alarm goes to state INSUFFICIENT_DATA.

delete ()

describe_history (*start_date=None, end_date=None, max_records=None, history_item_type=None, next_token=None*)

disable_actions ()

enable_actions ()

endElement (*name, value, connection*)

set_state (*value, reason, data=None*)

Temporarily sets the state of an alarm.

Parameters

- **value** (*str*) – OK | ALARM | INSUFFICIENT_DATA
- **reason** (*str*) – Reason alarm set (human readable).
- **data** (*str*) – Reason data (will be jsonified).

startElement (*name, attrs, connection*)

update ()

class boto.ec2.cloudwatch.alarm.**MetricAlarms** (*connection=None*)

Parses a list of MetricAlarms.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.118 CodeDeploy

4.118.1 boto.codedeploy

boto.codedeploy.connect_to_region (*region_name, **kw_params*)

boto.codedeploy.regions ()

Get all available regions for the AWS CodeDeploy service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.118.2 boto.coddeploy.layer1

class `boto.coddeploy.layer1.CodeDeployConnection` (**kwargs)

AWS CodeDeploy Overview This is the AWS CodeDeploy API Reference. This guide provides descriptions of the AWS CodeDeploy APIs. For additional information, see the '[AWS CodeDeploy User Guide](#)'. **Using the APIs** You can use the AWS CodeDeploy APIs to work with the following items:

- **Applications** , which are unique identifiers that AWS CodeDeploy uses to ensure that the correct combinations of revisions, deployment configurations, and deployment groups are being referenced during deployments. You can work with applications by calling `CreateApplication`, `DeleteApplication`, `GetApplication`, `ListApplications`, `BatchGetApplications`, and `UpdateApplication` to create, delete, and get information about applications, and to change information about an application, respectively.
- **Deployment configurations** , which are sets of deployment rules and deployment success and failure conditions that AWS CodeDeploy uses during deployments. You can work with deployment configurations by calling `CreateDeploymentConfig`, `DeleteDeploymentConfig`, `GetDeploymentConfig`, and `ListDeploymentConfigs` to create, delete, and get information about deployment configurations, respectively.
- **Deployment groups** , which represent groups of Amazon EC2 instances to which application revisions can be deployed. You can work with deployment groups by calling `CreateDeploymentGroup`, `DeleteDeploymentGroup`, `GetDeploymentGroup`, `ListDeploymentGroups`, and `UpdateDeploymentGroup` to create, delete, and get information about single and multiple deployment groups, and to change information about a deployment group, respectively.
- **Deployment instances** (also known simply as instances) , which represent Amazon EC2 instances to which application revisions are deployed. Deployment instances are identified by their Amazon EC2 tags or Auto Scaling group names. Deployment instances belong to deployment groups. You can work with deployment instances by calling `GetDeploymentInstance` and `ListDeploymentInstances` to get information about single and multiple deployment instances, respectively.
- **Deployments** , which represent the process of deploying revisions to deployment groups. You can work with deployments by calling `CreateDeployment`, `GetDeployment`, `ListDeployments`, `BatchGetDeployments`, and `StopDeployment` to create and get information about deployments, and to stop a deployment, respectively.
- **Application revisions** (also known simply as revisions) , which are archive files that are stored in Amazon S3 buckets or GitHub repositories. These revisions contain source content (such as source code, web pages, executable files, any deployment scripts, and similar) along with an Application Specification file (AppSpec file). (The AppSpec file is unique to AWS CodeDeploy; it defines a series of deployment actions that you want AWS CodeDeploy to execute.) An application revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. Application revisions are deployed to deployment groups. You can work with application revisions by calling `GetApplicationRevision`, `ListApplicationRevisions`, and `RegisterApplicationRevision` to get information about application revisions and to inform AWS CodeDeploy about an application revision, respectively.

APIVersion = '2014-10-06'

DefaultRegionEndpoint = 'coddeploy.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

ServiceName = 'coddeploy'

TargetPrefix = 'CodeDeploy_20141006'

batch_get_applications (*application_names=None*)

Gets information about one or more applications.

Parameters `application_names` (*list*) – A list of application names, with multiple application names separated by spaces.

batch_get_deployments (*deployment_ids=None*)

Gets information about one or more deployments.

Parameters `deployment_ids` (*list*) – A list of deployment IDs, with multiple deployment IDs separated by spaces.

create_application (*application_name*)

Creates a new application.

Parameters `application_name` (*string*) – The name of the application. This name must be unique within the AWS user account.

create_deployment (*application_name, deployment_group_name=None, revision=None, deployment_config_name=None, description=None, ignore_application_stop_failures=None*)

Deploys an application revision to the specified deployment group.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **deployment_group_name** (*string*) – The deployment group’s name.
- **revision** (*dict*) – The type of revision to deploy, along with information about the revision’s location.
- **deployment_config_name** (*string*) – The name of an existing deployment configuration within the AWS user account.

If not specified, the value configured in the deployment group will be used as the default. If the deployment group does not have a deployment configuration associated with it, then CodeDeployDefault.OneAtATime will be used by default.

Parameters

- **description** (*string*) – A comment about the deployment.
- **ignore_application_stop_failures** (*boolean*) – If set to true, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will not be considered to have failed to that instance at that point and will continue on to the BeforeInstall deployment lifecycle event.

If set to false or not specified, then if the deployment causes the ApplicationStop deployment lifecycle event to fail to a specific instance, the deployment will stop to that instance, and the deployment to that instance will be considered to have failed.

create_deployment_config (*deployment_config_name, minimum_healthy_hosts=None*)

Creates a new deployment configuration.

Parameters

- **deployment_config_name** (*string*) – The name of the deployment configuration to create.
- **minimum_healthy_hosts** (*dict*) – The minimum number of healthy instances that should be available at any time during the deployment. There are two parameters expected in the input: type and value.

The type parameter takes either of the following values:

- **HOST_COUNT:** The value parameter represents the minimum number of healthy instances, as an absolute value.

- **FLEET_PERCENT**: The value parameter represents the minimum number of healthy instances, as a percentage of the total number of instances in the deployment. If you specify FLEET_PERCENT, then at the start of the deployment AWS CodeDeploy converts the percentage to the equivalent number of instances and rounds fractional instances up.

The value parameter takes an integer.

For example, to set a minimum of 95% healthy instances, specify a type of FLEET_PERCENT and a value of 95.

```
create_deployment_group (application_name, deployment_group_name, deployment_config_name=None, ec_2_tag_filters=None, auto_scaling_groups=None, service_role_arn=None)
```

Creates a new deployment group for application revisions to be deployed to.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
 - **deployment_group_name** (*string*) – The name of an existing deployment group for the specified application.
 - **deployment_config_name** (*string*) – If specified, the deployment configuration name must be one of the predefined values, or it can be a custom deployment configuration:
- **CodeDeployDefault.AllAtOnce** deploys an application revision to up to all of the Amazon EC2 instances at once. The overall deployment succeeds if the application revision deploys to at least one of the instances. The overall deployment fails after the application revision fails to deploy to all of the instances. For example, for 9 instances, deploy to up to all 9 instances at once. The overall deployment succeeds if any of the 9 instances is successfully deployed to, and it fails if all 9 instances fail to be deployed to.
 - **CodeDeployDefault.HalfAtATime** deploys to up to half of the instances at a time (with fractions rounded down). The overall deployment succeeds if the application revision deploys to at least half of the instances (with fractions rounded up); otherwise, the deployment fails. For example, for 9 instances, deploy to up to 4 instances at a time. The overall deployment succeeds if 5 or more instances are successfully deployed to; otherwise, the deployment fails. Note that the deployment may successfully deploy to some instances, even if the overall deployment fails.
 - **CodeDeployDefault.OneAtATime** deploys the application revision to only one of the instances at a time. The overall deployment succeeds if the application revision deploys to all of the instances. The overall deployment fails after the application revision first fails to deploy to any one instance. For example, for 9 instances, deploy to one instance at a time. The overall deployment succeeds if all 9 instances are successfully deployed to, and it fails if any of one of the 9 instances fail to be deployed to. Note that the deployment may successfully deploy to some instances, even if the overall deployment fails. This is the default deployment configuration if a configuration isn't specified for either the deployment or the deployment group.

To create a custom deployment configuration, call the **create deployment configuration** operation.

Parameters

- **ec_2_tag_filters** (*list*) – The Amazon EC2 tags to filter on.
- **auto_scaling_groups** (*list*) – A list of associated Auto Scaling groups.
- **service_role_arn** (*string*) – A service role ARN that allows AWS CodeDeploy to act on the user's behalf when interacting with AWS services.

```
delete_application (application_name)
```

Deletes an application.

Parameters `application_name` (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.

delete_deployment_config (*deployment_config_name*)

Deletes a deployment configuration.

A deployment configuration cannot be deleted if it is currently in use. Also, predefined configurations cannot be deleted.

Parameters `deployment_config_name` (*string*) – The name of an existing deployment configuration within the AWS user account.

delete_deployment_group (*application_name, deployment_group_name*)

Deletes a deployment group.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **deployment_group_name** (*string*) – The name of an existing deployment group for the specified application.

get_application (*application_name*)

Gets information about an application.

Parameters `application_name` (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.

get_application_revision (*application_name, revision*)

Gets information about an application revision.

Parameters

- **application_name** (*string*) – The name of the application that corresponds to the revision.
- **revision** (*dict*) – Information about the application revision to get, including the revision's type and its location.

get_deployment (*deployment_id*)

Gets information about a deployment.

Parameters `deployment_id` (*string*) – An existing deployment ID within the AWS user account.

get_deployment_config (*deployment_config_name*)

Gets information about a deployment configuration.

Parameters `deployment_config_name` (*string*) – The name of an existing deployment configuration within the AWS user account.

get_deployment_group (*application_name, deployment_group_name*)

Gets information about a deployment group.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **deployment_group_name** (*string*) – The name of an existing deployment group for the specified application.

get_deployment_instance (*deployment_id, instance_id*)

Gets information about an Amazon EC2 instance as part of a deployment.

Parameters

- **deployment_id** (*string*) – The unique ID of a deployment.

- **instance_id** (*string*) – The unique ID of an Amazon EC2 instance in the deployment’s deployment group.

list_application_revisions (*application_name*, *sort_by=None*, *sort_order=None*,
s_3_bucket=None, *s_3_key_prefix=None*, *deployed=None*,
next_token=None)

Lists information about revisions for an application.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **sort_by** (*string*) – The column name to sort the list results by:
- **registerTime**: Sort the list results by when the revisions were registered with AWS CodeDeploy.
- **firstUsedTime**: Sort the list results by when the revisions were first used by in a deployment.
- **lastUsedTime**: Sort the list results by when the revisions were last used in a deployment.

If not specified or set to null, the results will be returned in an arbitrary order.

Parameters **sort_order** (*string*) – The order to sort the list results by:

- ascending: Sort the list results in ascending order.
- descending: Sort the list results in descending order.

If not specified, the results will be sorted in ascending order.

If set to null, the results will be sorted in an arbitrary order.

Parameters **s_3_bucket** (*string*) – A specific Amazon S3 bucket name to limit the search for revisions.

If set to null, then all of the user’s buckets will be searched.

Parameters

- **s_3_key_prefix** (*string*) – A specific key prefix for the set of Amazon S3 objects to limit the search for revisions.
- **deployed** (*string*) –

Whether to list revisions based on whether the revision is the target revision of an deployment group:

- **include**: List revisions that are target revisions of a deployment group.
- **exclude**: Do not list revisions that are target revisions of a deployment group.
- **ignore**: List all revisions, regardless of whether they are target revisions of a deployment group.

Parameters **next_token** (*string*) – An identifier that was returned from the previous list application revisions call, which can be used to return the next set of applications in the list.

list_applications (*next_token=None*)

Lists the applications registered within the AWS user account.

Parameters **next_token** (*string*) – An identifier that was returned from the previous list applications call, which can be used to return the next set of applications in the list.

list_deployment_configs (*next_token=None*)

Lists the deployment configurations within the AWS user account.

Parameters **next_token** (*string*) – An identifier that was returned from the previous list deployment configurations call, which can be used to return the next set of deployment configurations in the list.

list_deployment_groups (*application_name*, *next_token=None*)

Lists the deployment groups for an application registered within the AWS user account.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **next_token** (*string*) – An identifier that was returned from the previous list deployment groups call, which can be used to return the next set of deployment groups in the list.

list_deployment_instances (*deployment_id*, *next_token=None*, *instance_status_filter=None*)

Lists the Amazon EC2 instances for a deployment within the AWS user account.

Parameters

- **deployment_id** (*string*) – The unique ID of a deployment.
- **next_token** (*string*) – An identifier that was returned from the previous list deployment instances call, which can be used to return the next set of deployment instances in the list.
- **instance_status_filter** (*list*) –

A subset of instances to list, by status:

- **Pending: Include in the resulting list those instances with pending** deployments.
- **InProgress: Include in the resulting list those instances with in-** progress deployments.
- **Succeeded: Include in the resulting list those instances with** succeeded deployments.
- **Failed: Include in the resulting list those instances with failed** deployments.
- **Skipped: Include in the resulting list those instances with skipped** deployments.
- **Unknown: Include in the resulting list those instances with** deployments in an unknown state.

list_deployments (*application_name=None*, *deployment_group_name=None*, *include_only_statuses=None*, *create_time_range=None*, *next_token=None*)

Lists the deployments under a deployment group for an application registered within the AWS user account.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **deployment_group_name** (*string*) – The name of an existing deployment group for the specified application.
- **include_only_statuses** (*list*) – A subset of deployments to list, by status:

- **Created:** Include in the resulting list created deployments.
- **Queued:** Include in the resulting list queued deployments.
- **In Progress:** Include in the resulting list in-progress deployments.
- **Succeeded:** Include in the resulting list succeeded deployments.
- **Failed:** Include in the resulting list failed deployments.
- **Aborted:** Include in the resulting list aborted deployments.

Parameters

- **create_time_range** (*dict*) – A deployment creation start- and end-time range for returning a subset of the list of deployments.
- **next_token** (*string*) – An identifier that was returned from the previous list deployments call, which can be used to return the next set of deployments in the list.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

register_application_revision (*application_name, revision, description=None*)

Registers with AWS CodeDeploy a revision for the specified application.

Parameters

- **application_name** (*string*) – The name of an existing AWS CodeDeploy application within the AWS user account.
- **description** (*string*) – A comment about the revision.
- **revision** (*dict*) – Information about the application revision to register, including the revision’s type and its location.

stop_deployment (*deployment_id*)

Attempts to stop an ongoing deployment.

Parameters **deployment_id** (*string*) – The unique ID of a deployment.

update_application (*application_name=None, new_application_name=None*)

Changes an existing application’s name.

Parameters

- **application_name** (*string*) – The current name of the application that you want to change.
- **new_application_name** (*string*) – The new name that you want to change the application to.

update_deployment_group (*application_name, current_deployment_group_name, new_deployment_group_name=None, deployment_config_name=None, ec_2_tag_filters=None, auto_scaling_groups=None, service_role_arn=None*)

Changes information about an existing deployment group.

Parameters

- **application_name** (*string*) – The application name corresponding to the deployment group to update.
- **current_deployment_group_name** (*string*) – The current name of the existing deployment group.
- **new_deployment_group_name** (*string*) – The new name of the deployment group, if you want to change it.
- **deployment_config_name** (*string*) – The replacement deployment configuration name to use, if you want to change it.
- **ec_2_tag_filters** (*list*) – The replacement set of Amazon EC2 tags to filter on, if you want to change them.
- **auto_scaling_groups** (*list*) – The replacement list of Auto Scaling groups to be included in the deployment group, if you want to change them.
- **service_role_arn** (*string*) – A replacement service role’s ARN, if you want to change it.

4.118.3 boto.codedeploy.exceptions

```
exception boto.codedeploy.exceptions.ApplicationAlreadyExistsException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.ApplicationDoesNotExistException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.ApplicationLimitExceededException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.ApplicationNameRequiredException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.BucketNameFilterRequiredException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.DeploymentAlreadyCompletedException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.DeploymentConfigAlreadyExistsException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.DeploymentConfigDoesNotExistException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.DeploymentConfigInUseException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.codedeploy.exceptions.DeploymentConfigLimitExceededException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.coddeploy.exceptions.DeploymentConfigNameRequiredException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentDoesNotExistException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentGroupAlreadyExistsException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentGroupDoesNotExistException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentGroupLimitExceededException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentGroupNameRequiredException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentIdRequiredException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentLimitExceededException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DeploymentNotStartedException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.DescriptionTooLongException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InstanceDoesNotExistException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.coddeploy.exceptions.InstanceIdRequiredException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidApplicationNameException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidAutoScalingGroupException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidBucketNameFilterException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidDeployedStateFilterException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidDeploymentConfigNameException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidDeploymentGroupNameException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidDeploymentIdException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidDeploymentStatusException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidEC2TagException (status,      reason,  
                                                                    body=None, *args)  
exception boto.coddeploy.exceptions.InvalidInstanceStatusException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.coddeploy.exceptions.InvalidKeyPrefixFilterException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidMinimumHealthyHostValueException (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidNextTokenException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidOperationException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidRevisionException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidRoleException (status, reason,  
                                                                    body=None, *args)  
exception boto.coddeploy.exceptions.InvalidSortByException (status, reason,  
                                                                    body=None, *args)  
exception boto.coddeploy.exceptions.InvalidSortOrderException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.InvalidTimeRangeException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.RevisionDoesNotExistException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.RevisionRequiredException (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.coddeploy.exceptions.RoleRequiredException (status, reason,  
                                                                    body=None, *args)
```

4.119 Cognito Identity

4.119.1 boto.cognito.identity

`boto.cognito.identity.connect_to_region` (*region_name*, ***kw_params*)

`boto.cognito.identity.regions` ()

Get all available regions for the Amazon Cognito Identity service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.119.2 boto.cognito.identity.layer1

class `boto.cognito.identity.layer1.CognitoIdentityConnection` (***kwargs*)

Amazon Cognito Amazon Cognito is a web service that delivers scoped temporary credentials to mobile devices and other untrusted environments. Amazon Cognito uniquely identifies a device and supplies the user with a consistent identity over the lifetime of an application.

Using Amazon Cognito, you can enable authentication with one or more third-party identity providers (Facebook, Google, or Login with Amazon), and you can also choose to support unauthenticated access from your app. Cognito delivers a unique identifier for each user and acts as an OpenID token provider trusted by AWS Security Token Service (STS) to access temporary, limited- privilege AWS credentials.

To provide end-user credentials, first make an unsigned call to `GetId`. If the end user is authenticated with one of the supported identity providers, set the *Logins* map with the identity provider token. *GetId* returns a unique identifier for the user.

Next, make an unsigned call to `GetOpenIdToken`, which returns the OpenID token necessary to call STS and retrieve AWS credentials. This call expects the same *Logins* map as the *GetId* call, as well as the *IdentityID* originally returned by *GetId*. The token returned by *GetOpenIdToken* can be passed to the STS operation `'AssumeRoleWithWebIdentity'` to retrieve AWS credentials.

APIVersion = '2014-06-30'

DefaultRegionEndpoint = 'cognito-identity.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

ServiceName = 'CognitoIdentity'

TargetPrefix = 'AWSCognitoIdentityService'

create_identity_pool (*identity_pool_name*, *allow_unauthenticated_identities*, *supported_login_providers=None*, *developer_provider_name=None*, *open_id_connect_provider_ar_ns=None*)

Creates a new identity pool. The identity pool is a store of user identity information that is specific to your AWS account. The limit on identity pools is 60 per account.

Parameters

- **identity_pool_name** (*string*) – A string that you provide.
- **allow_unauthenticated_identities** (*boolean*) – TRUE if the identity pool supports unauthenticated logins.
- **supported_login_providers** (*map*) – Optional key:value pairs mapping provider names to provider app IDs.
- **developer_provider_name** (*string*) – The “domain” by which Cognito will refer to your users. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider. For the *DeveloperProviderName*, you can use letters as well as period (.), underscore (_), and dash (-).

Once you have set a developer provider name, you cannot change it. Please take care in setting this parameter.

Parameters **open_id_connect_provider_ar_ns** (*list*) –

delete_identity_pool (*identity_pool_id*)

Deletes a user pool. Once a pool is deleted, users will not be able to authenticate with the pool.

Parameters `identity_pool_id` (*string*) – An identity pool ID in the format REGION:GUID.

describe_identity_pool (*identity_pool_id*)

Gets details about a particular identity pool, including the pool name, ID description, creation date, and current number of users.

Parameters `identity_pool_id` (*string*) – An identity pool ID in the format REGION:GUID.

get_id (*account_id*, *identity_pool_id*, *logins=None*)

Generates (or retrieves) a Cognito ID. Supplying multiple logins will create an implicit linked account.

Parameters

- **account_id** (*string*) – A standard AWS account ID (9+ digits).
- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **logins** (*map*) – A set of optional name-value pairs that map provider names to provider tokens.

The available provider names for *Logins* are as follows:

- Facebook: *graph.facebook.com*
- Google: *accounts.google.com*
- Amazon: *www.amazon.com*

get_open_id_token (*identity_id*, *logins=None*)

Gets an OpenID token, using a known Cognito ID. This known Cognito ID is returned by `GetId`. You can optionally add additional logins for the identity. Supplying multiple logins creates an implicit link.

The OpenId token is valid for 15 minutes.

Parameters

- **identity_id** (*string*) – A unique identifier in the format REGION:GUID.
- **logins** (*map*) – A set of optional name-value pairs that map provider names to provider tokens.

get_open_id_token_for_developer_identity (*identity_pool_id*, *logins*, *identity_id=None*, *token_duration=None*)

Registers (or retrieves) a Cognito *IdentityId* and an OpenID Connect token for a user authenticated by your backend authentication process. Supplying multiple logins will create an implicit linked account. You can only specify one developer provider as part of the *Logins* map, which is linked to the identity pool. The developer provider is the “domain” by which Cognito will refer to your users.

You can use *GetOpenIdTokenForDeveloperIdentity* to create a new identity and to link new logins (that is, user credentials issued by a public provider or developer provider) to an existing identity. When you want to create a new identity, the *IdentityId* should be null. When you want to associate a new login with an existing authenticated/unauthenticated identity, you can do so by providing the existing *IdentityId*. This API will create the identity in the specified *IdentityPoolId*.

Parameters

- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **identity_id** (*string*) – A unique identifier in the format REGION:GUID.
- **logins** (*map*) – A set of optional name-value pairs that map provider names to provider tokens. Each name-value pair represents a user from a public provider or developer provider. If the user is from a developer provider, the name-value pair will follow the syntax “*developer_provider_name*”: “*developer_user_identifer*”. The developer provider is the “domain” by which Cog-

nito will refer to your users; you provided this domain while creating/updating the identity pool. The developer user identifier is an identifier from your backend that uniquely identifies a user. When you create an identity pool, you can specify the supported logins.

- **token_duration** (*long*) – The expiration time of the token, in seconds. You can specify a custom expiration time for the token so that you can cache it. If you don't provide an expiration time, the token is valid for 15 minutes. You can exchange the token with Amazon STS for temporary AWS credentials, which are valid for a maximum of one hour. The maximum token duration you can set is 24 hours. You should take care in setting the expiration time for a token, as there are significant security implications: an attacker could use a leaked token to access your AWS resources for the token's duration.

list_identities (*identity_pool_id*, *max_results*, *next_token=None*)

Lists the identities in a pool.

Parameters

- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **max_results** (*integer*) – The maximum number of identities to return.
- **next_token** (*string*) – A pagination token.

list_identity_pools (*max_results*, *next_token=None*)

Lists all of the Cognito identity pools registered for your account.

Parameters

- **max_results** (*integer*) – The maximum number of identities to return.
- **next_token** (*string*) – A pagination token.

lookup_developer_identity (*identity_pool_id*, *identity_id=None*, *developer_user_identifier=None*, *max_results=None*, *next_token=None*)

Retrieves the *IdentityID* associated with a *DeveloperUserIdentifier* or the list of *DeveloperUserIdentifier*'s associated with an *IdentityId* for an existing identity. Either *IdentityID* or *DeveloperUserIdentifier* must not be null. If you supply only one of these values, the other value will be searched in the database and returned as a part of the response. If you supply both, *DeveloperUserIdentifier* will be matched against *IdentityID*. If the values are verified against the database, the response returns both values and is the same as the request. Otherwise a *ResourceConflictException* is thrown.

Parameters

- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **identity_id** (*string*) – A unique identifier in the format REGION:GUID.
- **developer_user_identifier** (*string*) – A unique ID used by your backend authentication process to identify a user. Typically, a developer identity provider would issue many developer user identifiers, in keeping with the number of users.
- **max_results** (*integer*) – The maximum number of identities to return.
- **next_token** (*string*) – A pagination token. The first call you make will have *NextToken* set to null. After that the service will return *NextToken* values as needed. For example, let's say you make a request with *MaxResults* set to 10, and there are 20 matches in the database. The service will return a pagination token

as a part of the response. This token can be used to call the API again and get results starting from the 11th match.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

merge_developer_identities (*source_user_identifier, destination_user_identifier, developer_provider_name, identity_pool_id*)

Merges two users having different *IdentityId*'s, existing in the same identity pool, and identified by the same developer provider. You can use this action to request that discrete users be merged and identified as a single user in the Cognito environment. Cognito associates the given source user (*SourceUserIdentifier*) with the *IdentityId* of the *DestinationUserIdentifier*. Only developer-authenticated users can be merged. If the users to be merged are associated with the same public provider, but as two different users, an exception will be thrown.

Parameters

- **source_user_identifier** (*string*) – User identifier for the source user. The value should be a *DeveloperUserIdentifier*.
- **destination_user_identifier** (*string*) – User identifier for the destination user. The value should be a *DeveloperUserIdentifier*.
- **developer_provider_name** (*string*) – The “domain” by which Cognito will refer to your users. This is a (pseudo) domain name that you provide while creating an identity pool. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider. For the *DeveloperProviderName*, you can use letters as well as period (.), underscore (_), and dash (-).
- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.

unlink_developer_identity (*identity_id, identity_pool_id, developer_provider_name, developer_user_identifier*)

Unlinks a *DeveloperUserIdentifier* from an existing identity. Unlinked developer users will be considered new identities next time they are seen. If, for a given Cognito identity, you remove all federated identities as well as the developer user identifier, the Cognito identity becomes inaccessible.

Parameters

- **identity_id** (*string*) – A unique identifier in the format REGION:GUID.
- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **developer_provider_name** (*string*) – The “domain” by which Cognito will refer to your users.
- **developer_user_identifier** (*string*) – A unique ID used by your backend authentication process to identify a user.

unlink_identity (*identity_id, logins, logins_to_remove*)

Unlinks a federated identity from an existing account. Unlinked logins will be considered new identities next time they are seen. Removing the last linked login will make this identity inaccessible.

Parameters

- **identity_id** (*string*) – A unique identifier in the format REGION:GUID.
- **logins** (*map*) – A set of optional name-value pairs that map provider names to provider tokens.
- **logins_to_remove** (*list*) – Provider names to unlink from this identity.

update_identity_pool (*identity_pool_id*, *identity_pool_name*, *allow_unauthenticated_identities*, *supported_login_providers=None*, *developer_provider_name=None*, *open_id_connect_provider_ar_ns=None*)

Updates a user pool.

Parameters

- **identity_pool_id** (*string*) – An identity pool ID in the format REGION:GUID.
- **identity_pool_name** (*string*) – A string that you provide.
- **allow_unauthenticated_identities** (*boolean*) – TRUE if the identity pool supports unauthenticated logins.
- **supported_login_providers** (*map*) – Optional key:value pairs mapping provider names to provider app IDs.
- **developer_provider_name** (*string*) – The “domain” by which Cognito will refer to your users.
- **open_id_connect_provider_ar_ns** (*list*) –

4.119.3 boto.cognito.identity.exceptions

exception boto.cognito.identity.exceptions.**DeveloperUserAlreadyRegisteredException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**InternalErrorException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**InvalidParameterException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**LimitExceededException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**NotAuthorizedException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**ResourceConflictException** (*status*, *reason*, *body=None*, **args*)

exception boto.cognito.identity.exceptions.**ResourceNotFoundException** (*status*, *reason*, *body=None*, **args*)

exception `boto.cognito.identity.exceptions.TooManyRequestsException` (*status*,
reason,
body=None,
**args*)

4.120 Cognito Sync

4.120.1 boto.cognito.sync

`boto.cognito.sync.connect_to_region` (*region_name*, ***kw_params*)

`boto.cognito.sync.regions` ()

Get all available regions for the Amazon Cognito Sync service.

Return type `list`

Returns A list of `boto.regioninfo.RegionInfo`

4.120.2 boto.cognito.sync.layer1

class `boto.cognito.sync.layer1.CognitoSyncConnection` (***kwargs*)

Amazon Cognito Sync provides an AWS service and client library that enable cross-device syncing of application-related user data. High-level client libraries are available for both iOS and Android. You can use these libraries to persist data locally so that it's available even if the device is offline. Developer credentials don't need to be stored on the mobile device to access the service. You can use Amazon Cognito to obtain a normalized user ID and credentials. User data is persisted in a dataset that can store up to 1 MB of key-value pairs, and you can have up to 20 datasets per user identity.

With Amazon Cognito Sync, the data stored for each identity is accessible only to credentials assigned to that identity. In order to use the Cognito Sync service, you need to make API calls using credentials retrieved with **'Amazon Cognito Identity service'**.

APIVersion = '2014-06-30'

DefaultRegionEndpoint = 'cognito-sync.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

delete_dataset (*identity_pool_id*, *identity_id*, *dataset_name*)

Deletes the specific dataset. The dataset will be deleted permanently, and the action can't be undone. Datasets that this dataset was merged with will no longer report the merge. Any consequent operation on this dataset will result in a `ResourceNotFoundException`.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **dataset_name** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).

describe_dataset (*identity_pool_id*, *identity_id*, *dataset_name*)

Gets metadata about a dataset by identity and dataset name. The credentials used to make this API call need to have access to the identity data. With Amazon Cognito Sync, each identity has access only to its own data. You should use Amazon Cognito Identity service to retrieve the credentials necessary to make this API call.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **dataset_name** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).

describe_identity_pool_usage (*identity_pool_id*)

Gets usage details (for example, data storage) about a particular identity pool.

Parameters **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

describe_identity_usage (*identity_pool_id*, *identity_id*)

Gets usage information for an identity, including number of datasets and data usage.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.

get_identity_pool_configuration (*identity_pool_id*)

Gets the configuration settings of an identity pool.

Parameters **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. This is the ID of the pool for which to return a configuration.

list_datasets (*identity_pool_id*, *identity_id*, *next_token=None*, *max_results=None*)

Lists datasets for an identity. The credentials used to make this API call need to have access to the identity data. With Amazon Cognito Sync, each identity has access only to its own data. You should use Amazon Cognito Identity service to retrieve the credentials necessary to make this API call.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **next_token** (*string*) – A pagination token for obtaining the next page of results.
- **max_results** (*integer*) – The maximum number of results to be returned.

list_identity_pool_usage (*next_token=None, max_results=None*)

Gets a list of identity pools registered with Cognito.

Parameters

- **next_token** (*string*) – A pagination token for obtaining the next page of results.
- **max_results** (*integer*) – The maximum number of results to be returned.

list_records (*identity_pool_id, identity_id, dataset_name, last_sync_count=None, next_token=None, max_results=None, sync_session_token=None*)

Gets paginated records, optionally changed after a particular sync count for a dataset and identity. The credentials used to make this API call need to have access to the identity data. With Amazon Cognito Sync, each identity has access only to its own data. You should use Amazon Cognito Identity service to retrieve the credentials necessary to make this API call.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **dataset_name** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, ‘_’ (underscore), ‘-’ (dash), and ‘.’ (dot).
- **last_sync_count** (*long*) – The last server sync count for this record.
- **next_token** (*string*) – A pagination token for obtaining the next page of results.
- **max_results** (*integer*) – The maximum number of results to be returned.
- **sync_session_token** (*string*) – A token containing a session ID, identity ID, and expiration.

make_request (*verb, resource, headers=None, data=”, expected_status=None, params=None*)

Makes a request to the server, with stock multiple-retry logic.

register_device (*identity_pool_id, identity_id, platform, token*)

Registers a device to receive push sync notifications.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. Here, the ID of the pool that the identity belongs to.
- **identity_id** (*string*) – The unique ID for this identity.
- **platform** (*string*) – The SNS platform type (e.g. GCM, SDM, APNS, APNS_SANDBOX).
- **token** (*string*) – The push token.

set_identity_pool_configuration (*identity_pool_id, push_sync=None*)

Sets the necessary configuration for push sync.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. This is the ID of the pool to modify.

- **push_sync** (*dict*) – Configuration options to be applied to the identity pool.

subscribe_to_dataset (*identity_pool_id, identity_id, dataset_name, device_id*)

Subscribes to receive notifications when a dataset is modified by another device.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. The ID of the pool to which the identity belongs.
- **identity_id** (*string*) – Unique ID for this identity.
- **dataset_name** (*string*) – The name of the dataset to subscribe to.
- **device_id** (*string*) – The unique ID generated for this device by Cognito.

unsubscribe_from_dataset (*identity_pool_id, identity_id, dataset_name, device_id*)

Unsubscribe from receiving notifications when a dataset is modified by another device.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. The ID of the pool to which this identity belongs.
- **identity_id** (*string*) – Unique ID for this identity.
- **dataset_name** (*string*) – The name of the dataset from which to unsubscribe.
- **device_id** (*string*) – The unique ID generated for this device by Cognito.

update_records (*identity_pool_id, identity_id, dataset_name, sync_session_token, device_id=None, record_patches=None, client_context=None*)

Posts updates to records and add and delete records for a dataset and user. The credentials used to make this API call need to have access to the identity data. With Amazon Cognito Sync, each identity has access only to its own data. You should use Amazon Cognito Identity service to retrieve the credentials necessary to make this API call.

Parameters

- **identity_pool_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **identity_id** (*string*) – A name-spaced GUID (for example, us-east-1:23EC4050-6AEA-7089-A2DD-08002EXAMPLE) created by Amazon Cognito. GUID generation is unique within a region.
- **dataset_name** (*string*) – A string of up to 128 characters. Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot).
- **device_id** (*string*) – The unique ID generated for this device by Cognito.
- **record_patches** (*list*) – A list of patch operations.
- **sync_session_token** (*string*) – The SyncSession token returned by a previous call to ListRecords for this dataset and identity.
- **client_context** (*string*) – Intended to supply a device ID that will populate the *lastModifiedBy* field referenced in other methods. The *ClientContext* field is not yet implemented.

4.120.3 boto.cognito.sync.exceptions

exception boto.cognito.sync.exceptions.**InternalErrorException** (*status*, *reason*,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**InvalidConfigurationException** (*status*,
reason,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**InvalidParameterException** (*status*,
reason,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**LimitExceededException** (*status*, *reason*,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**NotAuthorizedException** (*status*, *reason*,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**ResourceConflictException** (*status*,
reason,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**ResourceNotFoundException** (*status*,
reason,
body=None,
**args*)

exception boto.cognito.sync.exceptions.**TooManyRequestsException** (*status*, *reason*,
body=None,
**args*)

4.121 Config

4.121.1 boto.configservice

boto.configservice.**connect_to_region** (*region_name*, ***kw_params*)

boto.configservice.**regions** ()

Get all available regions for the AWS Config service.

Return type list

Returns A list of boto.regioninfo.RegionInfo

4.121.2 boto.configservice.layer1

class boto.configservice.layer1.**ConfigServiceConnection** (***kwargs*)

AWS Config provides a way to keep track of the configurations of all the AWS resources associated with your AWS account. You can use AWS Config to get the current and historical configurations of each AWS resource and also to get information about the relationship between the resources. An AWS resource can be an Amazon Compute Cloud (Amazon EC2) instance, an Elastic Block Store (EBS) volume, an Elastic network

Interface (ENI), or a security group. For a complete list of resources currently supported by AWS Config, see [‘Supported AWS Resources’](#).

You can access and manage AWS Config through the AWS Management Console, the AWS Command Line Interface (AWS CLI), the AWS Config API, or the AWS SDKs for AWS Config

This reference guide contains documentation for the AWS Config API and the AWS CLI commands that you can use to manage AWS Config.

The AWS Config API uses the Signature Version 4 protocol for signing requests. For more information about how to sign a request with this protocol, see [‘Signature Version 4 Signing Process’](#).

For detailed information about AWS Config features and their associated actions or commands, as well as how to work with AWS Management Console, see [‘What Is AWS Config?’](#) in the AWS Config Developer Guide .

APIVersion = '2014-11-12'

DefaultRegionEndpoint = 'config.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.JSONResponseError*

ServiceName = 'ConfigService'

TargetPrefix = 'StarlingDoveService'

delete_delivery_channel (*delivery_channel_name*)

Deletes the specified delivery channel.

The delivery channel cannot be deleted if it is the only delivery channel and the configuration recorder is still running. To delete the delivery channel, stop the running configuration recorder using the StopConfigurationRecorder action.

Parameters **delivery_channel_name** (*string*) – The name of the delivery channel to delete.

deliver_config_snapshot (*delivery_channel_name*)

Schedules delivery of a configuration snapshot to the Amazon S3 bucket in the specified delivery channel. After the delivery has started, AWS Config sends following notifications using an Amazon SNS topic that you have specified.

- Notification of starting the delivery.
- Notification of delivery completed, if the delivery was successfully completed.
- Notification of delivery failure, if the delivery failed to complete.

Parameters **delivery_channel_name** (*string*) – The name of the delivery channel through which the snapshot is delivered.

describe_configuration_recorder_status (*configuration_recorder_names=None*)

Returns the current status of the specified configuration recorder. If a configuration recorder is not specified, this action returns the status of all configuration recorder associated with the account.

Parameters **configuration_recorder_names** (*list*) – The name(s) of the configuration recorder. If the name is not specified, the action returns the current status of all the configuration recorders associated with the account.

describe_configuration_recorders (*configuration_recorder_names=None*)

Returns the name of one or more specified configuration recorders. If the recorder name is not specified, this action returns the names of all the configuration recorders associated with the account.

Parameters **configuration_recorder_names** (*list*) – A list of configuration recorder names.

describe_delivery_channel_status (*delivery_channel_names=None*)

Returns the current status of the specified delivery channel. If a delivery channel is not specified, this action returns the current status of all delivery channels associated with the account.

Parameters **delivery_channel_names** (*list*) – A list of delivery channel names.

describe_delivery_channels (*delivery_channel_names=None*)

Returns details about the specified delivery channel. If a delivery channel is not specified, this action returns the details of all delivery channels associated with the account.

Parameters **delivery_channel_names** (*list*) – A list of delivery channel names.

get_resource_config_history (*resource_type, resource_id, later_time=None, earlier_time=None, chronological_order=None, limit=None, next_token=None*)

Returns a list of configuration items for the specified resource. The list contains details about each state of the resource during the specified time interval. You can specify a *limit* on the number of results returned on the page. If a limit is specified, a *nextToken* is returned as part of the result that you can use to continue this request.

Parameters

- **resource_type** (*string*) – The resource type.
- **resource_id** (*string*) – The ID of the resource (for example., *sg-xxxxxx*).
- **later_time** (*timestamp*) – The time stamp that indicates a later time. If not specified, current time is taken.
- **earlier_time** (*timestamp*) – The time stamp that indicates an earlier time. If not specified, the action returns paginated results that contain configuration items that start from when the first configuration item was recorded.
- **chronological_order** (*string*) – The chronological order for configuration items listed. By default the results are listed in reverse chronological order.
- **limit** (*integer*) – The maximum number of configuration items returned in each page. The default is 10. You cannot specify a limit greater than 100.
- **next_token** (*string*) – An optional parameter used for pagination of the results.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

put_configuration_recorder (*configuration_recorder*)

Creates a new configuration recorder to record the resource configurations.

You can use this action to change the role (*roleARN*) of an existing recorder. To change the role, call the action on the existing configuration recorder and specify a role.

Parameters **configuration_recorder** (*dict*) – The configuration recorder object that records each configuration change made to the resources. The format should follow:

```
{'name': 'myrecorder', 'roleARN': 'arn:aws:iam::123456789012:role/trusted-aws-config'}
```

put_delivery_channel (*delivery_channel*)

Creates a new delivery channel object to deliver the configuration information to an Amazon S3 bucket, and to an Amazon SNS topic.

You can use this action to change the Amazon S3 bucket or an Amazon SNS topic of the existing delivery channel. To change the Amazon S3 bucket or an Amazon SNS topic, call this action and specify the changed values for the S3 bucket and the SNS topic. If you specify a different value for either the S3 bucket or the SNS topic, this action will keep the existing value for the parameter that is not changed.

Parameters `delivery_channel` (*dict*) – The configuration delivery channel object that delivers the configuration information to an Amazon S3 bucket, and to an Amazon SNS topic.

start_configuration_recorder (*configuration_recorder_name*)

Starts recording configurations of all the resources associated with the account.

You must have created at least one delivery channel to successfully start the configuration recorder.

Parameters `configuration_recorder_name` (*string*) – The name of the recorder object that records each configuration change made to the resources.

stop_configuration_recorder (*configuration_recorder_name*)

Stops recording configurations of all the resources associated with the account.

Parameters `configuration_recorder_name` (*string*) – The name of the recorder object that records each configuration change made to the resources.

4.121.3 boto.configservice.exceptions

exception `boto.configservice.exceptions.InsufficientDeliveryPolicyException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidConfigurationRecorderNameException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidDeliveryChannelNameException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidLimitException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidNextTokenException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidRoleException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidS3KeyPrefixException` (*status*, *reason*, *body=None*, **args*)

exception `boto.configservice.exceptions.InvalidSNSTopicARNException` (*status*, *reason*, *body=None*, **args*)

```
exception boto.configservice.exceptions.InvalidTimeRangeException (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.configservice.exceptions.LastDeliveryChannelDeleteFailedException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.MaxNumberOfConfigurationRecordersExceededException
```

```
exception boto.configservice.exceptions.MaxNumberOfDeliveryChannelsExceededException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=N  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.NoAvailableConfigurationRecorderException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.NoAvailableDeliveryChannelException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.NoRunningConfigurationRecorderException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.NoSuchBucketException (status,    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.configservice.exceptions.NoSuchConfigurationRecorderException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

```
exception boto.configservice.exceptions.NoSuchDeliveryChannelException (status,  
                                                                                      rea-  
                                                                                      son,  
                                                                                      body=None,  
                                                                                      *args)
```

exception boto.configservice.exceptions.**ResourceNotDiscoveredException** (*status*,
reason,
body=None,
**args*)

exception boto.configservice.exceptions.**ValidationException** (*status*, *reason*,
body=None, **args*)

4.122 contrib

4.122.1 boto.contrib

4.122.2 boto.contrib.ymlmessage

This module was contributed by Chris Moyer. It provides a subclass of the SQS Message class that supports YAML as the body of the message.

This module requires the yaml module.

class boto.contrib.ymlmessage.**YAMLMessage** (*queue=None*, *body=""*, *xml_attrs=None*)

The YAMLMessage class provides a YAML compatible message. Encoding and decoding are handled automatically.

Access this message data like such:

```
m.data = [ 1, 2, 3] m.data[0] # Returns 1
```

This depends on the PyYAML package

get_body ()

set_body (*body*)

Override the current body for this object, using decoded format.

4.123 Data Pipeline

4.123.1 boto.datapipeline

boto.datapipeline.**connect_to_region** (*region_name*, ***kw_params*)

boto.datapipeline.**regions** ()

Get all available regions for the AWS Datapipeline service.

Return type list

Returns A list of boto.regioninfo.RegionInfo

4.123.2 boto.datapipeline.layer1

class boto.datapipeline.layer1.**DataPipelineConnection** (***kwargs*)

This is the AWS Data Pipeline API Reference . This guide provides descriptions and samples of the AWS Data Pipeline API.

AWS Data Pipeline is a web service that configures and manages a data-driven workflow called a pipeline. AWS Data Pipeline handles the details of scheduling and ensuring that data dependencies are met so your application can focus on processing the data.

The AWS Data Pipeline API implements two main sets of functionality. The first set of actions configure the pipeline in the web service. You call these actions to create a pipeline and define data sources, schedules, dependencies, and the transforms to be performed on the data.

The second set of actions are used by a task runner application that calls the AWS Data Pipeline API to receive the next task ready for processing. The logic for performing the task, such as querying the data, running data analysis, or converting the data from one format to another, is contained within the task runner. The task runner performs the task assigned to it by the web service, reporting progress to the web service as it does so. When the task is done, the task runner reports the final success or failure of the task to the web service.

AWS Data Pipeline provides an open-source implementation of a task runner called AWS Data Pipeline Task Runner. AWS Data Pipeline Task Runner provides logic for common data management scenarios, such as performing database queries and running data analysis using Amazon Elastic MapReduce (Amazon EMR). You can use AWS Data Pipeline Task Runner as your task runner, or you can write your own task runner to provide custom data management.

The AWS Data Pipeline API uses the Signature Version 4 protocol for signing requests. For more information about how to sign a request with this protocol, see [‘Signature Version 4 Signing Process’](#). In the code examples in this reference, the Signature Version 4 Request parameters are represented as AuthParams.

```
APIVersion = '2012-10-29'
```

```
DefaultRegionEndpoint = 'datapipeline.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
ResponseError
```

```
alias of boto.exception.JSONResponseError
```

```
ServiceName = 'DataPipeline'
```

```
TargetPrefix = 'DataPipeline'
```

```
activate_pipeline (pipeline_id)
```

```
Validates a pipeline and initiates processing. If the pipeline does not pass validation, activation fails.
```

```
Call this action to start processing pipeline tasks of a pipeline you’ve created using the CreatePipeline and PutPipelineDefinition actions. A pipeline cannot be modified after it has been successfully activated.
```

```
Parameters pipeline_id (string) – The identifier of the pipeline to activate.
```

```
create_pipeline (name, unique_id, description=None)
```

```
Creates a new empty pipeline. When this action succeeds, you can then use the PutPipelineDefinition action to populate the pipeline.
```

```
Parameters
```

- **name** (*string*) – The name of the new pipeline. You can use the same name for multiple pipelines associated with your AWS account, because AWS Data Pipeline assigns each new pipeline a unique pipeline identifier.
- **unique_id** (*string*) – A unique identifier that you specify. This identifier is not the same as the pipeline identifier assigned by AWS Data Pipeline. You are responsible for defining the format and ensuring the uniqueness of this identifier. You use this parameter to ensure idempotency during repeated calls to CreatePipeline. For example, if the first call to CreatePipeline does not return a clear success, you can pass in the same unique identifier and pipeline name combination on a subsequent call to CreatePipeline. CreatePipeline ensures that if a pipeline already exists with the same name and unique identifier, a new pipeline

will not be created. Instead, you'll receive the pipeline identifier from the previous attempt. The uniqueness of the name and unique identifier combination is scoped to the AWS account or IAM user credentials.

- **description** (*string*) – The description of the new pipeline.

delete_pipeline (*pipeline_id*)

Permanently deletes a pipeline, its pipeline definition and its run history. You cannot query or restore a deleted pipeline. AWS Data Pipeline will attempt to cancel instances associated with the pipeline that are currently being processed by task runners. Deleting a pipeline cannot be undone.

To temporarily pause a pipeline instead of deleting it, call `SetStatus` with the status set to `Pause` on individual components. Components that are paused by `SetStatus` can be resumed.

Parameters **pipeline_id** (*string*) – The identifier of the pipeline to be deleted.

describe_objects (*object_ids, pipeline_id, marker=None, evaluate_expressions=None*)

Returns the object definitions for a set of objects associated with the pipeline. Object definitions are composed of a set of fields that define the properties of the object.

Parameters

- **pipeline_id** (*string*) – Identifier of the pipeline that contains the object definitions.
- **object_ids** (*list*) – Identifiers of the pipeline objects that contain the definitions to be described. You can pass as many as 25 identifiers in a single call to `DescribeObjects`.
- **evaluate_expressions** (*boolean*) – Indicates whether any expressions in the object should be evaluated when the object descriptions are returned.
- **marker** (*string*) – The starting point for the results to be returned. The first time you call `DescribeObjects`, this value should be empty. As long as the action returns `HasMoreResults` as `True`, you can call `DescribeObjects` again and pass the marker value from the response to retrieve the next set of results.

describe_pipelines (*pipeline_ids*)

Retrieve metadata about one or more pipelines. The information retrieved includes the name of the pipeline, the pipeline identifier, its current state, and the user account that owns the pipeline. Using account credentials, you can retrieve metadata about pipelines that you or your IAM users have created. If you are using an IAM user account, you can retrieve metadata about only those pipelines you have read permission for.

To retrieve the full pipeline definition instead of metadata about the pipeline, call the `GetPipelineDefinition` action.

Parameters **pipeline_ids** (*list*) – Identifiers of the pipelines to describe. You can pass as many as 25 identifiers in a single call to `DescribePipelines`. You can obtain pipeline identifiers by calling `ListPipelines`.

evaluate_expression (*pipeline_id, expression, object_id*)

Evaluates a string in the context of a specified object. A task runner can use this action to evaluate SQL queries stored in Amazon S3.

Parameters

- **pipeline_id** (*string*) – The identifier of the pipeline.
- **object_id** (*string*) – The identifier of the object.
- **expression** (*string*) – The expression to evaluate.

get_pipeline_definition (*pipeline_id, version=None*)

Returns the definition of the specified pipeline. You can call `GetPipelineDefinition` to retrieve the pipeline

definition you provided using `PutPipelineDefinition`.

Parameters

- **pipeline_id** (*string*) – The identifier of the pipeline.
- **version** (*string*) – The version of the pipeline definition to retrieve. This parameter accepts the values *latest* (default) and *active*. Where *latest* indicates the last definition saved to the pipeline and *active* indicates the last definition of the pipeline that was activated.

list_pipelines (*marker=None*)

Returns a list of pipeline identifiers for all active pipelines. Identifiers are returned only for pipelines you have permission to access.

Parameters **marker** (*string*) – The starting point for the results to be returned. The first time you call `ListPipelines`, this value should be empty. As long as the action returns *HasMoreResults* as *True*, you can call `ListPipelines` again and pass the marker value from the response to retrieve the next set of results.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

poll_for_task (*worker_group, hostname=None, instance_identity=None*)

Task runners call this action to receive a task to perform from AWS Data Pipeline. The task runner specifies which tasks it can perform by setting a value for the `workerGroup` parameter of the `PollForTask` call. The task returned by `PollForTask` may come from any of the pipelines that match the `workerGroup` value passed in by the task runner and that was launched using the IAM user credentials specified by the task runner.

If tasks are ready in the work queue, `PollForTask` returns a response immediately. If no tasks are available in the queue, `PollForTask` uses long-polling and holds on to a poll connection for up to a 90 seconds during which time the first newly scheduled task is handed to the task runner. To accommodate this, set the socket timeout in your task runner to 90 seconds. The task runner should not call `PollForTask` again on the same `workerGroup` until it receives a response, and this may take up to 90 seconds.

Parameters

- **worker_group** (*string*) – Indicates the type of task the task runner is configured to accept and process. The worker group is set as a field on objects in the pipeline when they are created. You can only specify a single value for *workerGroup* in the call to `PollForTask`. There are no wildcard values permitted in *workerGroup*, the string must be an exact, case-sensitive, match.
- **hostname** (*string*) – The public DNS name of the calling task runner.
- **instance_identity** (*dict*) – Identity information for the Amazon EC2 instance that is hosting the task runner. You can get this value by calling the URI, <http://169.254.169.254/latest/meta-data/instance-id>, from the EC2 instance. For more information, go to **‘Instance Metadata’** in the Amazon Elastic Compute Cloud User Guide. Passing in this value proves that your task runner is running on an EC2 instance, and ensures the proper AWS Data Pipeline service charges are applied to your pipeline.

put_pipeline_definition (*pipeline_objects, pipeline_id*)

Adds tasks, schedules, and preconditions that control the behavior of the pipeline. You can use `PutPipelineDefinition` to populate a new pipeline or to update an existing pipeline that has not yet been activated.

`PutPipelineDefinition` also validates the configuration as it adds it to the pipeline. Changes to the pipeline are saved unless one of the following three validation errors exists in the pipeline.

1. An object is missing a name or identifier field.

2. A string or reference field is empty.
3. The number of objects in the pipeline exceeds the maximum allowed objects.

Pipeline object definitions are passed to the PutPipelineDefinition action and returned by the GetPipelineDefinition action.

Parameters

- **pipeline_id** (*string*) – The identifier of the pipeline to be configured.
- **pipeline_objects** (*list*) – The objects that define the pipeline. These will overwrite the existing pipeline definition.

query_objects (*pipeline_id, sphere, marker=None, query=None, limit=None*)

Queries a pipeline for the names of objects that match a specified set of conditions.

The objects returned by QueryObjects are paginated and then filtered by the value you set for query. This means the action may return an empty result set with a value set for marker. If *HasMoreResults* is set to *True*, you should continue to call QueryObjects, passing in the returned value for marker, until *HasMoreResults* returns *False*.

Parameters

- **pipeline_id** (*string*) – Identifier of the pipeline to be queried for object names.
- **query** (*dict*) – Query that defines the objects to be returned. The Query object can contain a maximum of ten selectors. The conditions in the query are limited to top-level String fields in the object. These filters can be applied to components, instances, and attempts.
- **sphere** (*string*) – Specifies whether the query applies to components or instances. Allowable values: *COMPONENT, INSTANCE, ATTEMPT*.
- **marker** (*string*) – The starting point for the results to be returned. The first time you call QueryObjects, this value should be empty. As long as the action returns *HasMoreResults* as *True*, you can call QueryObjects again and pass the marker value from the response to retrieve the next set of results.
- **limit** (*integer*) – Specifies the maximum number of object names that QueryObjects will return in a single call. The default value is 100.

report_task_progress (*task_id*)

Updates the AWS Data Pipeline service on the progress of the calling task runner. When the task runner is assigned a task, it should call ReportTaskProgress to acknowledge that it has the task within 2 minutes. If the web service does not receive this acknowledgement within the 2 minute window, it will assign the task in a subsequent PollForTask call. After this initial acknowledgement, the task runner only needs to report progress every 15 minutes to maintain its ownership of the task. You can change this reporting time from 15 minutes by specifying a *reportProgressTimeout* field in your pipeline. If a task runner does not report its status after 5 minutes, AWS Data Pipeline will assume that the task runner is unable to process the task and will reassign the task in a subsequent response to PollForTask. task runners should call ReportTaskProgress every 60 seconds.

Parameters **task_id** (*string*) – Identifier of the task assigned to the task runner. This value is provided in the TaskObject that the service returns with the response for the PollForTask action.

report_task_runner_heartbeat (*taskrunner_id, worker_group=None, hostname=None*)

Task runners call ReportTaskRunnerHeartbeat every 15 minutes to indicate that they are operational. In the case of AWS Data Pipeline Task Runner launched on a resource managed by AWS Data Pipeline, the web service can use this call to detect when the task runner application has failed and restart a new instance.

Parameters

- **taskrunner_id** (*string*) – The identifier of the task runner. This value should be unique across your AWS account. In the case of AWS Data Pipeline Task Runner launched on a resource managed by AWS Data Pipeline, the web service provides a unique identifier when it launches the application. If you have written a custom task runner, you should assign a unique identifier for the task runner.
- **worker_group** (*string*) – Indicates the type of task the task runner is configured to accept and process. The worker group is set as a field on objects in the pipeline when they are created. You can only specify a single value for *workerGroup* in the call to `ReportTaskRunnerHeartbeat`. There are no wildcard values permitted in *workerGroup*, the string must be an exact, case-sensitive, match.
- **hostname** (*string*) – The public DNS name of the calling task runner.

set_status (*object_ids*, *status*, *pipeline_id*)

Requests that the status of an array of physical or logical pipeline objects be updated in the pipeline. This update may not occur immediately, but is eventually consistent. The status that can be set depends on the type of object.

Parameters

- **pipeline_id** (*string*) – Identifies the pipeline that contains the objects.
- **object_ids** (*list*) – Identifies an array of objects. The corresponding objects can be either physical or components, but not a mix of both types.
- **status** (*string*) – Specifies the status to be set on all the objects in *objectIds*. For components, this can be either *PAUSE* or *RESUME*. For instances, this can be either *CANCEL*, *RERUN*, or *MARK_FINISHED*.

set_task_status (*task_id*, *task_status*, *error_id=None*, *error_message=None*, *error_stack_trace=None*)

Notifies AWS Data Pipeline that a task is completed and provides information about the final status. The task runner calls this action regardless of whether the task was successful. The task runner does not need to call `SetTaskStatus` for tasks that are canceled by the web service during a call to `ReportTaskProgress`.

Parameters

- **task_id** (*string*) – Identifies the task assigned to the task runner. This value is set in the `TaskObject` that is returned by the `PollForTask` action.
- **task_status** (*string*) – If *FINISHED*, the task successfully completed. If *FAILED* the task ended unsuccessfully. The *FALSE* value is used by preconditions.
- **error_id** (*string*) – If an error occurred during the task, this value specifies an id value that represents the error. This value is set on the physical attempt object. It is used to display error information to the user. It should not start with string “**Service_**” which is reserved by the system.
- **error_message** (*string*) – If an error occurred during the task, this value specifies a text description of the error. This value is set on the physical attempt object. It is used to display error information to the user. The web service does not parse this value.
- **error_stack_trace** (*string*) – If an error occurred during the task, this value specifies the stack trace associated with the error. This value is set on the physical attempt object. It is used to display error information to the user. The web service does not parse this value.

validate_pipeline_definition (*pipeline_objects*, *pipeline_id*)

Tests the pipeline definition with a set of validation checks to ensure that it is well formed and can run without error.

Parameters

- **pipeline_id** (*string*) – Identifies the pipeline whose definition is to be validated.
- **pipeline_objects** (*list*) – A list of objects that define the pipeline changes to validate against the pipeline.

4.123.3 boto.datapipeline.exceptions

exception `boto.datapipeline.exceptions.InternalServiceError` (*status*, *reason*, *body=None*, **args*)

exception `boto.datapipeline.exceptions.InvalidRequestException` (*status*, *reason*, *body=None*, **args*)

exception `boto.datapipeline.exceptions.PipelineDeletedException` (*status*, *reason*, *body=None*, **args*)

exception `boto.datapipeline.exceptions.PipelineNotFoundException` (*status*, *reason*, *body=None*, **args*)

exception `boto.datapipeline.exceptions.TaskNotFoundException` (*status*, *reason*, *body=None*, **args*)

4.124 DynamoDB

4.124.1 boto.dynamodb

`boto.dynamodb.connect_to_region` (*region_name*, ***kw_params*)

`boto.dynamodb.regions` ()

Get all available regions for the Amazon DynamoDB service.

Return type `list`

Returns A list of `boto.regioninfo.RegionInfo`

4.124.2 boto.dynamodb.layer1

class `boto.dynamodb.layer1.Layer1` (*aws_access_key_id=None*, *aws_secret_access_key=None*, *is_secure=True*, *port=None*, *proxy=None*, *proxy_port=None*, *debug=0*, *security_token=None*, *region=None*, *validate_certs=True*, *validate_checksums=True*, *profile_name=None*)

This is the lowest-level interface to DynamoDB. Methods at this layer map directly to API requests and parameters to the methods are either simple, scalar values or they are the Python equivalent of the JSON input as defined in the DynamoDB Developer's Guide. All responses are direct decoding of the JSON response bodies to Python data structures via the `json` or `simplejson` modules.

Variables `throughput_exceeded_events` – An integer variable that keeps a running total of the number of `ThroughputExceeded` responses this connection has received from Amazon DynamoDB.

`ConditionalCheckFailedError = 'ConditionalCheckFailedException'`

The error response returned when a conditional check fails

`DefaultRegionName = 'us-east-1'`

The default region name for DynamoDB API.

`NumberRetries = 10`

The number of times an error is retried.

`ResponseError`

alias of `boto.exception.DynamoDBResponseError`

`ServiceName = 'DynamoDB'`

The name of the Service

`SessionExpiredError = 'com.amazon.coral.service#ExpiredTokenException'`

The error response returned when session token has expired

`ThruputError = 'ProvisionedThroughputExceededException'`

The error response returned when provisioned throughput is exceeded

`ValidationError = 'ValidationException'`

The error response returned when an item is invalid in some way

`Version = '20111205'`

DynamoDB API version.

`batch_get_item` (*request_items*, *object_hook=None*)

Return a set of attributes for a multiple items in multiple tables using their primary keys.

Parameters *request_items* (*dict*) – A Python version of the `RequestItems` data structure defined by DynamoDB.

`batch_write_item` (*request_items*, *object_hook=None*)

This operation enables you to put or delete several items across multiple tables in a single API call.

Parameters *request_items* (*dict*) – A Python version of the `RequestItems` data structure defined by DynamoDB.

`create_table` (*table_name*, *schema*, *provisioned_throughput*)

Add a new table to your account. The table name must be unique among those associated with the account issuing the request. This request triggers an asynchronous workflow to begin creating the table. When the workflow is complete, the state of the table will be `ACTIVE`.

Parameters

- **`table_name`** (*str*) – The name of the table to create.
- **`schema`** (*dict*) – A Python version of the `KeySchema` data structure as defined by DynamoDB
- **`provisioned_throughput`** (*dict*) – A Python version of the `ProvisionedThroughput` data structure defined by DynamoDB.

`delete_item` (*table_name*, *key*, *expected=None*, *return_values=None*, *object_hook=None*)

Delete an item and all of its attributes by primary key. You can perform a conditional delete by specifying an expected rule.

Parameters

- **`table_name`** (*str*) – The name of the table containing the item.
- **`key`** (*dict*) – A Python version of the `Key` data structure defined by DynamoDB.

- **expected** (*dict*) – A Python version of the Expected data structure defined by DynamoDB.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

delete_table (*table_name*)

Deletes the table and all of it's data. After this request the table will be in the DELETING state until DynamoDB completes the delete operation.

Parameters **table_name** (*str*) – The name of the table to delete.

describe_table (*table_name*)

Returns information about the table including current state of the table, primary key schema and when the table was created.

Parameters **table_name** (*str*) – The name of the table to describe.

get_item (*table_name, key, attributes_to_get=None, consistent_read=False, object_hook=None*)

Return a set of attributes for an item that matches the supplied key.

Parameters

- **table_name** (*str*) – The name of the table containing the item.
- **key** (*dict*) – A Python version of the Key data structure defined by DynamoDB.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.

list_tables (*limit=None, start_table=None*)

Returns a dictionary of results. The dictionary contains a **TableNames** key whose value is a list of the table names. The dictionary could also contain a **LastEvaluatedTableName** key whose value would be the last table name returned if the complete list of table names was not returned. This value would then be passed as the *start_table* parameter on a subsequent call to this method.

Parameters

- **limit** (*int*) – The maximum number of tables to return.
- **start_table** (*str*) – The name of the table that starts the list. If you ran a previous *list_tables* and not all results were returned, the response dict would include a *LastEvaluatedTableName* attribute. Use that value here to continue the listing.

make_request (*action, body=", object_hook=None*)

Raises `DynamoDBExpiredTokenError` if the security token expires.

put_item (*table_name, item, expected=None, return_values=None, object_hook=None*)

Create a new item or replace an old item with a new item (including all attributes). If an item already exists in the specified table with the same primary key, the new item will completely replace the old item. You can perform a conditional put by specifying an expected rule.

Parameters

- **table_name** (*str*) – The name of the table in which to put the item.
- **item** (*dict*) – A Python version of the Item data structure defined by DynamoDB.

- **expected** (*dict*) – A Python version of the Expected data structure defined by DynamoDB.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

query (*table_name*, *hash_key_value*, *range_key_conditions=None*, *attributes_to_get=None*, *limit=None*, *consistent_read=False*, *scan_index_forward=True*, *exclusive_start_key=None*, *object_hook=None*, *count=False*)

Perform a query of DynamoDB. This version is currently punting and expecting you to provide a full and correct JSON body which is passed as is to DynamoDB.

Parameters

- **table_name** (*str*) – The name of the table to query.
- **key** – A DynamoDB-style HashKeyValue.
- **range_key_conditions** (*dict*) – A Python version of the RangeKeyConditions data structure.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **limit** (*int*) – The maximum number of items to return.
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Query operation, even if the operation has no matching items for the assigned filter.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **scan_index_forward** (*bool*) – Specified forward or backward traversal of the index. Default is forward (True).
- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the LastEvaluatedKey in that query.

scan (*table_name*, *scan_filter=None*, *attributes_to_get=None*, *limit=None*, *exclusive_start_key=None*, *object_hook=None*, *count=False*)

Perform a scan of DynamoDB. This version is currently punting and expecting you to provide a full and correct JSON body which is passed as is to DynamoDB.

Parameters

- **table_name** (*str*) – The name of the table to scan.
- **scan_filter** (*dict*) – A Python version of the ScanFilter data structure.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **limit** (*int*) – The maximum number of items to evaluate.
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Scan operation, even if the operation has no matching items for the assigned filter.

- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the LastEvaluatedKey in that query.

update_item(*table_name, key, attribute_updates, expected=None, return_values=None, object_hook=None*)

Edits an existing item's attributes. You can perform a conditional update (insert a new attribute name-value pair if it doesn't exist, or replace an existing name-value pair if it has certain expected attribute values).

Parameters

- **table_name** (*str*) – The name of the table.
- **key** (*dict*) – A Python version of the Key data structure defined by DynamoDB which identifies the item to be updated.
- **attribute_updates** (*dict*) – A Python version of the AttributeUpdates data structure defined by DynamoDB.
- **expected** (*dict*) – A Python version of the Expected data structure defined by DynamoDB.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

update_table(*table_name, provisioned_throughput*)

Updates the provisioned throughput for a given table.

Parameters

- **table_name** (*str*) – The name of the table to update.
- **provisioned_throughput** (*dict*) – A Python version of the ProvisionedThroughput data structure defined by DynamoDB.

4.124.3 boto.dynamodb.layer2

```
class boto.dynamodb.layer2.Layer2(aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, debug=0, security_token=None, region=None, validate_certs=True, dynamizer=<class 'boto.dynamodb.types.LossyFloatDynamizer'>, profile_name=None)
```

batch_get_item(*batch_list*)

Return a set of attributes for a multiple items in multiple tables using their primary keys.

Parameters **batch_list** (*boto.dynamodb.batch.BatchList*) – A BatchList object which consists of a list of `boto.dynamodb.batch.Batch` objects. Each Batch object contains the information about one batch of objects that you wish to retrieve in this request.

batch_write_item(*batch_list*)

Performs multiple Puts and Deletes in one batch.

Parameters **batch_list** (*boto.dynamodb.batch.BatchWriteList*) – A BatchWriteList object which consists of a list of `boto.dynamodb.batch.BatchWrite` objects. Each Batch object contains the information about one batch of objects that you wish to put or delete.

build_key_from_values (*schema, hash_key, range_key=None*)

Build a Key structure to be used for accessing items in Amazon DynamoDB. This method takes the supplied hash_key and optional range_key and validates them against the schema. If there is a mismatch, a TypeError is raised. Otherwise, a Python dict version of a Amazon DynamoDB Key data structure is returned.

Parameters

- **hash_key** (*int|float|str|unicode|Binary*) – The hash key of the item you are looking for. The type of the hash key should match the type defined in the schema.
- **range_key** (*int|float|str|unicode|Binary*) – The range key of the item you are looking for. This should be supplied only if the schema requires a range key. The type of the range key should match the type defined in the schema.

create_schema (*hash_key_name, hash_key_proto_value, range_key_name=None, range_key_proto_value=None*)

Create a Schema object used when creating a Table.

Parameters

- **hash_key_name** (*str*) – The name of the HashKey for the schema.
- **hash_key_proto_value** (*int|long|float|str|unicode|Binary*) – A sample or prototype of the type of value you want to use for the HashKey. Alternatively, you can also just pass in the Python type (e.g. int, float, etc.).
- **range_key_name** (*str*) – The name of the RangeKey for the schema. This parameter is optional.
- **range_key_proto_value** (*int|long|float|str|unicode|Binary*) – A sample or prototype of the type of value you want to use for the RangeKey. Alternatively, you can also pass in the Python type (e.g. int, float, etc.) This parameter is optional.

create_table (*name, schema, read_units, write_units*)

Create a new Amazon DynamoDB table.

Parameters

- **name** (*str*) – The name of the desired table.
- **schema** (*boto.dynamodb.schema.Schema*) – The Schema object that defines the schema used by this table.
- **read_units** (*int*) – The value for ReadCapacityUnits.
- **write_units** (*int*) – The value for WriteCapacityUnits.

Return type *boto.dynamodb.table.Table*

Returns A Table object representing the new Amazon DynamoDB table.

delete_item (*item, expected_value=None, return_values=None*)

Delete the item from Amazon DynamoDB.

Parameters

- **item** (*boto.dynamodb.item.Item*) – The Item to delete from Amazon DynamoDB.
- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.

- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

delete_table (*table*)

Delete this table and all items in it. After calling this the Table objects status attribute will be set to 'DELETING'.

Parameters **table** (*boto.dynamodb.table.Table*) – The Table object that is being deleted.

describe_table (*name*)

Retrieve information about an existing table.

Parameters **name** (*str*) – The name of the desired table.

dynamize_attribute_updates (*pending_updates*)

Convert a set of pending item updates into the structure required by Layer1.

dynamize_expected_value (*expected_value*)

Convert an expected_value parameter into the data structure required for Layer1.

dynamize_item (*item*)

dynamize_last_evaluated_key (*last_evaluated_key*)

Convert a last_evaluated_key parameter into the data structure required for Layer1.

dynamize_range_key_condition (*range_key_condition*)

Convert a layer2 range_key_condition parameter into the structure required by Layer1.

dynamize_scan_filter (*scan_filter*)

Convert a layer2 scan_filter parameter into the structure required by Layer1.

get_item (*table, hash_key, range_key=None, attributes_to_get=None, consistent_read=False, item_class=<class 'boto.dynamodb.item.Item'>*)

Retrieve an existing item from the table.

Parameters

- **table** (*boto.dynamodb.table.Table*) – The Table object from which the item is retrieved.
- **hash_key** (*int | long | float | str | unicode | Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key** (*int | long | float | str | unicode | Binary*) – The optional RangeKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of *boto.dynamodb.item.Item*

get_table (*name*)

Retrieve the Table object for an existing table.

Parameters **name** (*str*) – The name of the desired table.

Return type *boto.dynamodb.table.Table*

Returns A Table object representing the table.

list_tables (*limit=None*)

Return a list of the names of all tables associated with the current account and region.

Parameters **limit** (*int*) – The maximum number of tables to return.

lookup (*name*)

Retrieve the Table object for an existing table.

Parameters **name** (*str*) – The name of the desired table.

Return type *boto.dynamodb.table.Table*

Returns A Table object representing the table.

new_batch_list ()

Return a new, empty *boto.dynamodb.batch.BatchList* object.

new_batch_write_list ()

Return a new, empty *boto.dynamodb.batch.BatchWriteList* object.

put_item (*item, expected_value=None, return_values=None*)

Store a new item or completely replace an existing item in Amazon DynamoDB.

Parameters

- **item** (*boto.dynamodb.item.Item*) – The Item to write to Amazon DynamoDB.
- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

query (*table, hash_key, range_key_condition=None, attributes_to_get=None, request_limit=None, max_results=None, consistent_read=False, scan_index_forward=True, exclusive_start_key=None, item_class=<class 'boto.dynamodb.item.Item'>, count=False*)

Perform a query on the table.

Parameters

- **table** (*boto.dynamodb.table.Table*) – The Table object that is being queried.
- **hash_key** (*int/long/float/str/unicode/Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key_condition** (*boto.dynamodb.condition.Condition*) – A Condition object. Condition object can be one of the following types:
EQ|LE|LT|GE|GT|BEGINS_WITH|BETWEEN
The only condition which expects or will accept two values is 'BETWEEN', otherwise a single value should be passed to the Condition constructor.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.

- **request_limit** (*int*) – The maximum number of items to retrieve from Amazon DynamoDB on each request. You may want to set a specific `request_limit` based on the provisioned throughput of your table. The default behavior is to retrieve as many results as possible per request.
- **max_results** (*int*) – The maximum number of results that will be retrieved from Amazon DynamoDB in total. For example, if you only wanted to see the first 100 results from the query, regardless of how many were actually available, you could set `max_results` to 100 and the generator returned from the query method will only yield 100 results max.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **scan_index_forward** (*bool*) – Specified forward or backward traversal of the index. Default is forward (True).
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Query operation, even if the operation has no matching items for the assigned filter. If count is True, the actual items are not returned and the count is accessible as the `count` attribute of the returned object.
- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the `LastEvaluatedKey` in that query.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

Return type `boto.dynamodb.layer2.TableGenerator`

scan (*table, scan_filter=None, attributes_to_get=None, request_limit=None, max_results=None, exclusive_start_key=None, item_class=<class 'boto.dynamodb.item.Item'>, count=False*)
Perform a scan of DynamoDB.

Parameters

- **table** (`boto.dynamodb.table.Table`) – The Table object that is being scanned.
- **scan_filter** (*A dict*) – A dictionary where the key is the attribute name and the value is a `boto.dynamodb.condition.Condition` object. Valid Condition objects include:
 - EQ - equal (1)
 - NE - not equal (1)
 - LE - less than or equal (1)
 - LT - less than (1)
 - GE - greater than or equal (1)
 - GT - greater than (1)
 - NOT_NULL - attribute exists (0, use None)
 - NULL - attribute does not exist (0, use None)
 - CONTAINS - substring or value in list (1)
 - NOT_CONTAINS - absence of substring or value in list (1)
 - BEGINS_WITH - substring prefix (1)

- IN - exact match in list (N)
- BETWEEN - >= first value, <= second value (2)
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **request_limit** (*int*) – The maximum number of items to retrieve from Amazon DynamoDB on each request. You may want to set a specific `request_limit` based on the provisioned throughput of your table. The default behavior is to retrieve as many results as possible per request.
- **max_results** (*int*) – The maximum number of results that will be retrieved from Amazon DynamoDB in total. For example, if you only wanted to see the first 100 results from the query, regardless of how many were actually available, you could set `max_results` to 100 and the generator returned from the query method will only yield 100 results max.
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Scan operation, even if the operation has no matching items for the assigned filter. If `count` is True, the actual items are not returned and the count is accessible as the `count` attribute of the returned object.
- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the `LastEvaluatedKey` in that query.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

Return type `boto.dynamodb.layer2.TableGenerator`

table_from_schema (*name, schema*)

Create a Table object from a schema.

This method will create a Table object without making any API calls. If you know the name and schema of the table, you can use this method instead of `get_table`.

Example usage:

```
table = layer2.table_from_schema(
    'tablename',
    Schema.create(hash_key=('foo', 'N')))
```

Parameters

- **name** (*str*) – The name of the table.
- **schema** (`boto.dynamodb.schema.Schema`) – The schema associated with the table.

Return type `boto.dynamodb.table.Table`

Returns A Table object representing the table.

update_item (*item, expected_value=None, return_values=None*)

Commit pending item updates to Amazon DynamoDB.

Parameters

- **item** (`boto.dynamodb.item.Item`) – The Item to update in Amazon DynamoDB. It is expected that you would have called the `add_attribute`,

`put_attribute` and/or `delete_attribute` methods on this Item prior to calling this method. Those queued changes are what will be updated.

- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.
- **return_values** (*str*) – Controls the return of attribute name/value pairs before they were updated. Possible values are: None, 'ALL_OLD', 'UPDATED_OLD', 'ALL_NEW' or 'UPDATED_NEW'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned. If 'ALL_NEW' is specified, then all the attributes of the new version of the item are returned. If 'UPDATED_NEW' is specified, the new versions of only the updated attributes are returned.

update_throughput (*table, read_units, write_units*)

Update the ProvisionedThroughput for the Amazon DynamoDB Table.

Parameters

- **table** (*boto.dynamodb.table.Table*) – The Table object whose throughput is being updated.
- **read_units** (*int*) – The new value for ReadCapacityUnits.
- **write_units** (*int*) – The new value for WriteCapacityUnits.

use_decimals (*use_boolean=False*)

Use the `decimal.Decimal` type for encoding/decoding numeric types.

By default, ints/floats are used to represent numeric types ('N', 'NS') received from DynamoDB. Using the `Decimal` type is recommended to prevent loss of precision.

class `boto.dynamodb.layer2.TableGenerator` (*table, callable, remaining, item_class, kwargs*)

This is an object that wraps up the `table_generator` function. The only real reason to have this is that we want to be able to accumulate and return the `ConsumedCapacityUnits` element that is part of each response.

Variables

- **last_evaluated_key** – A sequence representing the key(s) of the item last evaluated, or None if no additional results are available.
- **remaining** – The remaining quantity of results requested.
- **table** – The table to which the call was made.

consumed_units

Returns a float representing the `ConsumedCapacityUnits` accumulated.

count

The total number of items retrieved thus far. This value changes with iteration and even when issuing a call with `count=True`, it is necessary to complete the iteration to assert an accurate count value.

next_response ()

Issue a call and return the result. You can invoke this method while iterating over the `TableGenerator` in order to skip to the next “page” of results.

response

The current response to the call from DynamoDB.

scanned_count

As above, but representing the total number of items scanned by DynamoDB, without regard to any filters.

4.124.4 boto.dynamodb.table

class `boto.dynamodb.table.Table` (*layer2, response*)

An Amazon DynamoDB table.

Variables

- **name** – The name of the table.
- **create_time** – The date and time that the table was created.
- **status** – The current status of the table. One of: ‘ACTIVE’, ‘UPDATING’, ‘DELETING’.
- **schema** – A `boto.dynamodb.schema.Schema` object representing the schema defined for the table.
- **item_count** – The number of items in the table. This value is set only when the Table object is created or refreshed and may not reflect the actual count.
- **size_bytes** – Total size of the specified table, in bytes. Amazon DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- **read_units** – The ReadCapacityUnits of the tables Provisioned Throughput.
- **write_units** – The WriteCapacityUnits of the tables Provisioned Throughput.
- **schema** – The Schema object associated with the table.

Parameters

- **layer2** (`boto.dynamodb.layer2.Layer2`) – A `Layer2` api object.
- **response** (`dict`) – The output of `boto.dynamodb.layer1.Layer1.describe_table`.

batch_get_item (*keys, attributes_to_get=None*)

Return a set of attributes for a multiple items from a single table using their primary keys. This abstraction removes the 100 Items per batch limitations as well as the “UnprocessedKeys” logic.

Parameters

- **keys** (`list`) – A list of scalar or tuple values. Each element in the list represents one Item to retrieve. If the schema for the table has both a HashKey and a RangeKey, each element in the list should be a tuple consisting of (`hash_key`, `range_key`). If the schema for the table contains only a HashKey, each element in the list should be a scalar value of the appropriate type for the table schema. NOTE: The maximum number of items that can be retrieved for a single operation is 100. Also, the number of items retrieved is constrained by a 1 MB size limit.
- **attributes_to_get** (`list`) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.

Returns A `TableBatchGenerator` (generator) object which will iterate over all results

Return type `boto.dynamodb.table.TableBatchGenerator`

classmethod **create_from_schema** (*layer2, name, schema*)

Create a `Table` object.

If you know the name and schema of your table, you can create a `Table` object without having to make any API calls (normally an API call is made to retrieve the schema of a table).

Example usage:


```
table = Table.create_from_schema(
    boto.connect_dynamodb(),
    'tablename',
    Schema.create(hash_key=('keyname', 'N')))
```

Parameters

- **layer2** (*boto.dynamodb.layer2.Layer2*) – A Layer2 api object.
- **name** (*str*) – The name of the table.
- **schema** (*boto.dynamodb.schema.Schema*) – The schema associated with the table.

Return type *boto.dynamodb.table.Table*

Returns A Table object representing the table.

create_time

delete()

Delete this table and all items in it. After calling this the Table objects status attribute will be set to 'DELETING'.

get_item (*hash_key*, *range_key=None*, *attributes_to_get=None*, *consistent_read=False*, *item_class=<class 'boto.dynamodb.item.Item'>*)

Retrieve an existing item from the table.

Parameters

- **hash_key** (*int|long|float|str|unicode|Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key** (*int|long|float|str|unicode|Binary*) – The optional RangeKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of *boto.dynamodb.item.Item*

has_item (*hash_key*, *range_key=None*, *consistent_read=False*)

Checks the table to see if the Item with the specified *hash_key* exists. This may save a tiny bit of time/bandwidth over a straight *get_item()* if you have no intention to touch the data that is returned, since this method specifically tells Amazon not to return anything but the Item's key.

Parameters

- **hash_key** (*int|long|float|str|unicode|Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key** (*int|long|float|str|unicode|Binary*) – The optional RangeKey of the requested item. The type of the value must match the type defined in the schema for the table.

- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.

Return type `bool`

Returns True if the Item exists, False if not.

item_count

lookup (*hash_key*, *range_key=None*, *attributes_to_get=None*, *consistent_read=False*, *item_class=<class 'boto.dynamodb.item.Item'>*)

Retrieve an existing item from the table.

Parameters

- **hash_key** (*int | long | float | str | unicode | Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key** (*int | long | float | str | unicode | Binary*) – The optional RangeKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

name

new_item (*hash_key=None*, *range_key=None*, *attrs=None*, *item_class=<class 'boto.dynamodb.item.Item'>*)

Return a new, unsaved Item which can later be PUT to Amazon DynamoDB.

This method has explicit (but optional) parameters for the `hash_key` and `range_key` values of the item. You can use these explicit parameters when calling the method, such as:

```
>>> my_item = my_table.new_item(hash_key='a', range_key=1,
                                attrs={'key1': 'val1', 'key2': 'val2'})
>>> my_item
{'u'bar': 1, u'foo': 'a', 'key1': 'val1', 'key2': 'val2'}
```

Or, if you prefer, you can simply put the `hash_key` and `range_key` in the `attrs` dictionary itself, like this:

```
>>> attrs = {'foo': 'a', 'bar': 1, 'key1': 'val1', 'key2': 'val2'}
>>> my_item = my_table.new_item(attrs=attrs)
>>> my_item
{'u'bar': 1, u'foo': 'a', 'key1': 'val1', 'key2': 'val2'}
```

The effect is the same.

Parameters

- **hash_key** (*int | long | float | str | unicode | Binary*) – The HashKey of the new item. The type of the value must match the type defined in the schema for the table.

- **range_key** (*int|long|float|str|unicode|Binary*) – The optional RangeKey of the new item. The type of the value must match the type defined in the schema for the table.
- **attrs** (*dict*) – A dictionary of key value pairs used to populate the new item.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

query (*hash_key, *args, **kw*)

Perform a query on the table.

Parameters

- **hash_key** (*int|long|float|str|unicode|Binary*) – The HashKey of the requested item. The type of the value must match the type defined in the schema for the table.
- **range_key_condition** (`boto.dynamodb.condition.Condition`) – A Condition object. Condition object can be one of the following types:
EQ|LE|LT|GE|GT|BEGINS_WITH|BETWEEN
The only condition which expects or will accept two values is ‘BETWEEN’, otherwise a single value should be passed to the Condition constructor.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **request_limit** (*int*) – The maximum number of items to retrieve from Amazon DynamoDB on each request. You may want to set a specific request_limit based on the provisioned throughput of your table. The default behavior is to retrieve as many results as possible per request.
- **max_results** (*int*) – The maximum number of results that will be retrieved from Amazon DynamoDB in total. For example, if you only wanted to see the first 100 results from the query, regardless of how many were actually available, you could set max_results to 100 and the generator returned from the query method will only yeild 100 results max.
- **consistent_read** (*bool*) – If True, a consistent read request is issued. Otherwise, an eventually consistent request is issued.
- **scan_index_forward** (*bool*) – Specified forward or backward traversal of the index. Default is forward (True).
- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the LastEvaluatedKey in that query.
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Query operation, even if the operation has no matching items for the assigned filter. If count is True, the actual items are not returned and the count is accessible as the count attribute of the returned object.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

read_units

refresh (*wait_for_active=False, retry_seconds=5*)

Refresh all of the fields of the Table object by calling the underlying DescribeTable request.

Parameters

- **wait_for_active** (*bool*) – If True, this command will not return until the table status, as returned from Amazon DynamoDB, is 'ACTIVE'.
- **retry_seconds** (*int*) – If wait_for_active is True, this parameter controls the number of seconds of delay between calls to update_table in Amazon DynamoDB. Default is 5 seconds.

scan (*args, **kw)

Scan through this table, this is a very long and expensive operation, and should be avoided if at all possible.

Parameters

- **scan_filter** (*A dict*) – A dictionary where the key is the attribute name and the value is a `boto.dynamodb.condition.Condition` object. Valid Condition objects include:
 - EQ - equal (1)
 - NE - not equal (1)
 - LE - less than or equal (1)
 - LT - less than (1)
 - GE - greater than or equal (1)
 - GT - greater than (1)
 - NOT_NULL - attribute exists (0, use None)
 - NULL - attribute does not exist (0, use None)
 - CONTAINS - substring or value in list (1)
 - NOT_CONTAINS - absence of substring or value in list (1)
 - BEGINS_WITH - substring prefix (1)
 - IN - exact match in list (N)
 - BETWEEN - >= first value, <= second value (2)
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **request_limit** (*int*) – The maximum number of items to retrieve from Amazon DynamoDB on each request. You may want to set a specific request_limit based on the provisioned throughput of your table. The default behavior is to retrieve as many results as possible per request.
- **max_results** (*int*) – The maximum number of results that will be retrieved from Amazon DynamoDB in total. For example, if you only wanted to see the first 100 results from the query, regardless of how many were actually available, you could set max_results to 100 and the generator returned from the query method will only yield 100 results max.
- **count** (*bool*) – If True, Amazon DynamoDB returns a total number of items for the Scan operation, even if the operation has no matching items for the assigned filter. If count is True, the actual items are not returned and the count is accessible as the `count` attribute of the returned object.

- **exclusive_start_key** (*list or tuple*) – Primary key of the item from which to continue an earlier query. This would be provided as the LastEvaluatedKey in that query.
- **item_class** (*Class*) – Allows you to override the class used to generate the items. This should be a subclass of `boto.dynamodb.item.Item`

Returns A TableGenerator (generator) object which will iterate over all results

Return type `boto.dynamodb.layer2.TableGenerator`

schema

size_bytes

status

update_from_response (*response*)

Update the state of the Table object based on the response data received from Amazon DynamoDB.

update_throughput (*read_units, write_units*)

Update the ProvisionedThroughput for the Amazon DynamoDB Table.

Parameters

- **read_units** (*int*) – The new value for ReadCapacityUnits.
- **write_units** (*int*) – The new value for WriteCapacityUnits.

write_units

class `boto.dynamodb.table.TableBatchGenerator` (*table, keys, attributes_to_get=None, consistent_read=False*)

A low-level generator used to page through results from batch_get_item operations.

Variables `consumed_units` – An integer that holds the number of ConsumedCapacityUnits accumulated thus far for this generator.

4.124.5 boto.dynamodb.schema

class `boto.dynamodb.schema.Schema` (*schema_dict*)

Represents a DynamoDB schema.

Variables

- **hash_key_name** – The name of the hash key of the schema.
- **hash_key_type** – The DynamoDB type specification for the hash key of the schema.
- **range_key_name** – The name of the range key of the schema or None if no range key is defined.
- **range_key_type** – The DynamoDB type specification for the range key of the schema or None if no range key is defined.
- **dict** – The underlying Python dictionary that needs to be passed to Layer1 methods.

classmethod `create` (*hash_key, range_key=None*)

Convenience method to create a schema object.

Example usage:

```
schema = Schema.create(hash_key=('foo', 'N'))
schema2 = Schema.create(hash_key=('foo', 'N'),
                        range_key=('bar', 'S'))
```

Parameters

- **hash_key** (*tuple*) – A tuple of (hash_key_name, hash_key_type)
- **hash_key** – A tuple of (range_key_name, range_key_type)

dict

hash_key_name

hash_key_type

range_key_name

range_key_type

4.124.6 boto.dynamodb.item

class boto.dynamodb.item.**Item** (*table, hash_key=None, range_key=None, attrs=None*)
An item in Amazon DynamoDB.

Variables

- **hash_key** – The HashKey of this item.
- **range_key** – The RangeKey of this item or None if no RangeKey is defined.
- **hash_key_name** – The name of the HashKey associated with this item.
- **range_key_name** – The name of the RangeKey associated with this item.
- **table** – The Table this item belongs to.

add_attribute (*attr_name, attr_value*)

Queue the addition of an attribute to an item in DynamoDB. This will eventually result in an UpdateItem request being issued with an update action of ADD when the save method is called.

Parameters

- **attr_name** (*str*) – Name of the attribute you want to alter.
- **attr_value** (*int | long | float | set*) – Value which is to be added to the attribute.

delete (*expected_value=None, return_values=None*)

Delete the item from DynamoDB.

Parameters

- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

delete_attribute (*attr_name, attr_value=None*)

Queue the deletion of an attribute from an item in DynamoDB. This call will result in a UpdateItem request being issued with update action of DELETE when the save method is called.

Parameters

- **attr_name** (*str*) – Name of the attribute you want to alter.

- **attr_value** (*set*) – A set of values to be removed from the attribute. This parameter is optional. If None, the whole attribute is removed from the item.

hash_key

hash_key_name

put (*expected_value=None, return_values=None*)

Store a new item or completely replace an existing item in Amazon DynamoDB.

Parameters

- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.
- **return_values** (*str*) – Controls the return of attribute name-value pairs before then were changed. Possible values are: None or 'ALL_OLD'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned.

put_attribute (*attr_name, attr_value*)

Queue the putting of an attribute to an item in DynamoDB. This call will result in an UpdateItem request being issued with the update action of PUT when the save method is called.

Parameters

- **attr_name** (*str*) – Name of the attribute you want to alter.
- **attr_value** (*int | long | float | str | set*) – New value of the attribute.

range_key

range_key_name

save (*expected_value=None, return_values=None*)

Commits pending updates to Amazon DynamoDB.

Parameters

- **expected_value** (*dict*) – A dictionary of name/value pairs that you expect. This dictionary should have name/value pairs where the name is the name of the attribute and the value is either the value you are expecting or False if you expect the attribute not to exist.
- **return_values** (*str*) – Controls the return of attribute name/value pairs before they were updated. Possible values are: None, 'ALL_OLD', 'UPDATED_OLD', 'ALL_NEW' or 'UPDATED_NEW'. If 'ALL_OLD' is specified and the item is overwritten, the content of the old item is returned. If 'ALL_NEW' is specified, then all the attributes of the new version of the item are returned. If 'UPDATED_NEW' is specified, the new versions of only the updated attributes are returned.

4.124.7 boto.dynamodb.batch

class boto.dynamodb.batch.**Batch** (*table, keys, attributes_to_get=None, consistent_read=False*)

Used to construct a BatchGet request.

Variables

- **table** – The Table object from which the item is retrieved.
- **keys** – A list of scalar or tuple values. Each element in the list represents one Item to retrieve. If the schema for the table has both a HashKey and a RangeKey, each element

in the list should be a tuple consisting of (hash_key, range_key). If the schema for the table contains only a HashKey, each element in the list should be a scalar value of the appropriate type for the table schema. NOTE: The maximum number of items that can be retrieved for a single operation is 100. Also, the number of items retrieved is constrained by a 1 MB size limit.

- **attributes_to_get** – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.
- **consistent_read** – Specify whether or not to use a consistent read. Defaults to False.

to_dict ()

Convert the Batch object into the format required for Layer1.

class boto.dynamodb.batch.**BatchList** (layer2)

A subclass of a list object that contains a collection of *boto.dynamodb.batch.Batch* objects.

add_batch (table, keys, attributes_to_get=None, consistent_read=False)

Add a Batch to this BatchList.

Parameters

- **table** (*boto.dynamodb.table.Table*) – The Table object in which the items are contained.
- **keys** (*list*) – A list of scalar or tuple values. Each element in the list represents one Item to retrieve. If the schema for the table has both a HashKey and a RangeKey, each element in the list should be a tuple consisting of (hash_key, range_key). If the schema for the table contains only a HashKey, each element in the list should be a scalar value of the appropriate type for the table schema. NOTE: The maximum number of items that can be retrieved for a single operation is 100. Also, the number of items retrieved is constrained by a 1 MB size limit.
- **attributes_to_get** (*list*) – A list of attribute names. If supplied, only the specified attribute names will be returned. Otherwise, all attributes will be returned.

resubmit ()

Resubmit the batch to get the next result set. The request object is rebuild from scratch meaning that all batch added between `submit` and `resubmit` will be lost.

Note: This method is experimental and subject to changes in future releases

submit ()

to_dict ()

Convert a BatchList object into format required for Layer1.

class boto.dynamodb.batch.**BatchWrite** (table, puts=None, deletes=None)

Used to construct a BatchWrite request. Each BatchWrite object represents a collection of PutItem and DeleteItem requests for a single Table.

Variables

- **table** – The Table object from which the item is retrieved.
- **puts** – A list of *boto.dynamodb.item.Item* objects that you want to write to DynamoDB.
- **deletes** – A list of scalar or tuple values. Each element in the list represents one Item to delete. If the schema for the table has both a HashKey and a RangeKey, each element in the list should be a tuple consisting of (hash_key, range_key). If the schema

for the table contains only a HashKey, each element in the list should be a scalar value of the appropriate type for the table schema.

to_dict ()

Convert the Batch object into the format required for Layer1.

class boto.dynamodb.batch.**BatchWriteList** (*layer2*)

A subclass of a list object that contains a collection of *boto.dynamodb.batch.BatchWrite* objects.

add_batch (*table, puts=None, deletes=None*)

Add a BatchWrite to this BatchWriteList.

Parameters

- **table** (*boto.dynamodb.table.Table*) – The Table object in which the items are contained.
- **puts** (list of *boto.dynamodb.item.Item* objects) – A list of items that you want to write to DynamoDB.
- **deletes** (*A list*) – A list of scalar or tuple values. Each element in the list represents one Item to delete. If the schema for the table has both a HashKey and a RangeKey, each element in the list should be a tuple consisting of (hash_key, range_key). If the schema for the table contains only a HashKey, each element in the list should be a scalar value of the appropriate type for the table schema.

submit ()

to_dict ()

Convert a BatchWriteList object into format required for Layer1.

4.124.8 boto.dynamodb.types

Some utility functions to deal with mapping Amazon DynamoDB types to Python types and vice-versa.

class boto.dynamodb.types.**Binary** (*value*)

encode ()

class boto.dynamodb.types.**Dynamizer**

Control serialization/deserialization of types.

This class controls the encoding of python types to the format that is expected by the DynamoDB API, as well as taking DynamoDB types and constructing the appropriate python types.

If you want to customize this process, you can subclass this class and override the encoding/decoding of specific types. For example:

```
'foo'      (Python type)
 |
 v
encode('foo')
 |
 v
_encode_s('foo')
 |
 v
{'S': 'foo'} (Encoding sent to/received from DynamoDB)
 |
 v
```

(continues on next page)

```

decode({'S': 'foo'})
  |
  v
_decode_s({'S': 'foo'})
  |
  v
'foo'      (Python type)

```

decode (*attr*)

Takes the format returned by DynamoDB and constructs the appropriate python type.

encode (*attr*)

Encodes a python type to the format expected by DynamoDB.

class boto.dynamodb.types.LossyFloatDynamizer

Use float/int instead of Decimal for numeric types.

This class is provided for backwards compatibility. Instead of using Decimals for the 'N', 'NS' types it uses ints/floats.

This class is deprecated and its usage is not encouraged, as doing so may result in loss of precision. Use the *Dynamizer* class instead.

class boto.dynamodb.types.NonBooleanDynamizer

Casting boolean type to numeric types.

This class is provided for backward compatibility.

boto.dynamodb.types.**convert_binary** (*n*)

boto.dynamodb.types.**convert_num** (*s*)

boto.dynamodb.types.**dynamize_value** (*val*)

Take a scalar Python value and return a dict consisting of the Amazon DynamoDB type specification and the value that needs to be sent to Amazon DynamoDB. If the type of the value is not supported, raise a TypeError

boto.dynamodb.types.**float_to_decimal** (*f*)

boto.dynamodb.types.**get_dynamodb_type** (*val*, *use_boolean=True*)

Take a scalar Python value and return a string representing the corresponding Amazon DynamoDB type. If the value passed in is not a supported type, raise a TypeError.

boto.dynamodb.types.**is_binary** (*n*)

boto.dynamodb.types.**is_num** (*n*, *boolean_as_int=True*)

boto.dynamodb.types.**is_str** (*n*)

boto.dynamodb.types.**item_object_hook** (*dct*)

A custom object hook for use when decoding JSON item bodys. This hook will transform Amazon DynamoDB JSON responses to something that maps directly to native Python types.

boto.dynamodb.types.**serialize_num** (*val*)

Cast a number to a string and perform validation to ensure no loss of precision.

4.125 DynamoDB2

4.125.1 High-Level API

boto.dynamodb2.fields

class boto.dynamodb2.fields.**AllIndex** (*name*, *parts*)

An index signifying all fields should be in the index.

Example:

```
>>> AllIndex('MostRecentlyJoined', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ])
```

projection_type = 'ALL'

class boto.dynamodb2.fields.**BaseIndexField** (*name*, *parts*)

An abstract class for defining schema indexes.

Contains most of the core functionality for the index. Subclasses must define a `projection_type` to pass to DynamoDB.

definition ()

Returns the attribute definition structure DynamoDB expects.

Example:

```
>>> index.definition()
{
  'AttributeName': 'username',
  'AttributeType': 'S',
}
```

schema ()

Returns the schema structure DynamoDB expects.

Example:

```
>>> index.schema()
{
  'IndexName': 'LastNameIndex',
  'KeySchema': [
    {
      'AttributeName': 'username',
      'KeyType': 'HASH',
    },
  ],
  'Projection': {
    'ProjectionType': 'KEYS_ONLY',
  }
}
```

class boto.dynamodb2.fields.**BaseSchemaField** (*name*, *data_type*='S')

An abstract class for defining schema fields.

Contains most of the core functionality for the field. Subclasses must define an `attr_type` to pass to DynamoDB.

Creates a Python schema field, to represent the data to pass to DynamoDB.

Requires a `name` parameter, which should be a string name of the field.

Optionally accepts a `data_type` parameter, which should be a constant from `boto.dynamodb2.types`. (Default: `STRING`)

attr_type = None

definition()

Returns the attribute definition structure DynamoDB expects.

Example:

```
>>> field.definition()
{
  'AttributeName': 'username',
  'AttributeType': 'S',
}
```

schema()

Returns the schema structure DynamoDB expects.

Example:

```
>>> field.schema()
{
  'AttributeName': 'username',
  'KeyType': 'HASH',
}
```

class `boto.dynamodb2.fields.GlobalAllIndex(*args, **kwargs)`

An index signifying all fields should be in the index.

Example:

```
>>> GlobalAllIndex('MostRecentlyJoined', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ],
... throughput={
...     'read': 2,
...     'write': 1,
... })
```

projection_type = 'ALL'

class `boto.dynamodb2.fields.GlobalBaseIndexField(*args, **kwargs)`

An abstract class for defining global indexes.

Contains most of the core functionality for the index. Subclasses must define a `projection_type` to pass to DynamoDB.

schema()

Returns the schema structure DynamoDB expects.

Example:

```
>>> index.schema()
{
  'IndexName': 'LastNameIndex',
```

(continues on next page)

(continued from previous page)

```

'KeySchema': [
    {
        'AttributeName': 'username',
        'KeyType': 'HASH',
    },
],
'Projection': {
    'ProjectionType': 'KEYS_ONLY',
},
'ProvisionedThroughput': {
    'ReadCapacityUnits': 5,
    'WriteCapacityUnits': 5
}
}

```

```
throughput = {'read': 5, 'write': 5}
```

class boto.dynamodb2.fields.**GlobalIncludeIndex**(*args, **kwargs)

An index signifying only certain fields should be in the index.

Example:

```

>>> GlobalIncludeIndex('GenderIndex', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ],
... includes=['gender'],
... throughput={
...     'read': 2,
...     'write': 1,
... })

```

```
projection_type = 'INCLUDE'
```

schema()

Returns the schema structure DynamoDB expects.

Example:

```

>>> index.schema()
{
    'IndexName': 'LastNameIndex',
    'KeySchema': [
        {
            'AttributeName': 'username',
            'KeyType': 'HASH',
        },
    ],
    'Projection': {
        'ProjectionType': 'KEYS_ONLY',
    },
    'ProvisionedThroughput': {
        'ReadCapacityUnits': 5,
        'WriteCapacityUnits': 5
    }
}

```

class boto.dynamodb2.fields.**GlobalKeysOnlyIndex**(*args, **kwargs)

An index signifying only key fields should be in the index.

Example:

```
>>> GlobalKeysOnlyIndex('MostRecentlyJoined', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ],
... throughput={
...     'read': 2,
...     'write': 1,
... })
```

projection_type = 'KEYS_ONLY'

class boto.dynamodb2.fields.**HashKey**(name, data_type='S')

An field representing a hash key.

Example:

```
>>> from boto.dynamodb2.types import NUMBER
>>> HashKey('username')
>>> HashKey('date_joined', data_type=NUMBER)
```

Creates a Python schema field, to represent the data to pass to DynamoDB.

Requires a name parameter, which should be a string name of the field.

Optionally accepts a data_type parameter, which should be a constant from boto.dynamodb2.types. (Default: STRING)

attr_type = 'HASH'

class boto.dynamodb2.fields.**IncludeIndex**(*args, **kwargs)

An index signifying only certain fields should be in the index.

Example:

```
>>> IncludeIndex('GenderIndex', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ], includes=['gender'])
```

projection_type = 'INCLUDE'

schema()

Returns the schema structure DynamoDB expects.

Example:

```
>>> index.schema()
{
  'IndexName': 'LastNameIndex',
  'KeySchema': [
    {
      'AttributeName': 'username',
      'KeyType': 'HASH',
    },
  ],
  'Projection': {
    'ProjectionType': 'KEYS_ONLY',
```

(continues on next page)

(continued from previous page)

```

    }
}

```

class boto.dynamodb2.fields.**KeysOnlyIndex** (*name*, *parts*)
An index signifying only key fields should be in the index.

Example:

```

>>> KeysOnlyIndex('MostRecentlyJoined', parts=[
...     HashKey('username'),
...     RangeKey('date_joined')
... ])

```

projection_type = 'KEYS_ONLY'

class boto.dynamodb2.fields.**RangeKey** (*name*, *data_type*='S')
An field representing a range key.

Example:

```

>>> from boto.dynamodb2.types import NUMBER
>>> HashKey('username')
>>> HashKey('date_joined', data_type=NUMBER)

```

Creates a Python schema field, to represent the data to pass to DynamoDB.

Requires a *name* parameter, which should be a string name of the field.

Optionally accepts a *data_type* parameter, which should be a constant from `boto.dynamodb2.types`. (Default: `STRING`)

attr_type = 'RANGE'

boto.dynamodb2.items

class boto.dynamodb2.items.**Item** (*table*, *data*=None, *loaded*=False)
An object representing the item data within a DynamoDB table.

An item is largely schema-free, meaning it can contain any data. The only limitation is that it must have data for the fields in the `Table`'s schema.

This object presents a dictionary-like interface for accessing/storing data. It also tries to intelligently track how data has changed throughout the life of the instance, to be as efficient as possible about updates.

Empty items, or items that have no data, are considered falsey.

Constructs an (unsaved) `Item` instance.

To persist the data in DynamoDB, you'll need to call the `Item.save` (or `Item.partial_save`) on the instance.

Requires a *table* parameter, which should be a `Table` instance. This is required, as DynamoDB's API is focus around all operations being table-level. It's also for persisting schema around many objects.

Optionally accepts a *data* parameter, which should be a dictionary of the fields & values of the item. Alternatively, an `Item` instance may be provided from which to extract the data.

Optionally accepts a *loaded* parameter, which should be a boolean. True if it was preexisting data loaded from DynamoDB, False if it's new data from the user. Default is `False`.

Example:

```

>>> users = Table('users')
>>> user = Item(users, data={
...     'username': 'johndoe',
...     'first_name': 'John',
...     'date_joined': 1248061592,
... })

# Change existing data.
>>> user['first_name'] = 'Johann'
# Add more data.
>>> user['last_name'] = 'Doe'
# Delete data.
>>> del user['date_joined']

# Iterate over all the data.
>>> for field, val in user.items():
...     print "%s: %s" % (field, val)
username: johndoe
first_name: Johann
date_joined: 1248061592

```

build_expectations (*fields=None*)

Builds up a list of expectations to hand off to DynamoDB on save.

Largely internal.

delete ()

Deletes the item's data to DynamoDB.

Returns True on success.

Example:

```

# Buh-bye now.
>>> user.delete()

```

get (*key, default=None*)**get_keys** ()

Returns a Python-style dict of the keys/values.

Largely internal.

get_raw_keys ()

Returns a DynamoDB-style dict of the keys/values.

Largely internal.

items ()**keys** ()**load** (*data*)

This is only useful when being handed raw data from DynamoDB directly. If you have a Python data-structure already, use the `__init__` or manually set the data instead.

Largely internal, unless you know what you're doing or are trying to mix the low-level & high-level APIs.

mark_clean ()

Marks an `Item` instance as no longer needing to be saved.

Example:


```

>>> user.needs_save()
False
>>> user['first_name'] = 'Johann'
>>> user.needs_save()
True
>>> user.mark_clean()
>>> user.needs_save()
False

```

mark_dirty()

DEPRECATED: Marks an `Item` instance as needing to be saved.

This method is no longer necessary, as the state tracking on `Item` has been improved to automatically detect proper state.

needs_save (data=None)

Returns whether or not the data has changed on the `Item`.

Optionally accepts a `data` argument, which accepts the output from `self._determine_alterations()` if you've already called it. Typically unnecessary to do. Default is `None`.

Example:

```

>>> user.needs_save()
False
>>> user['first_name'] = 'Johann'
>>> user.needs_save()
True

```

partial_save()

Saves only the changed data to DynamoDB.

Extremely useful for high-volume/high-write data sets, this allows you to update only a handful of fields rather than having to push entire items. This prevents many accidental overwrite situations as well as saves on the amount of data to transfer over the wire.

Returns `True` on success, `False` if no save was performed or the write failed.

Example:

```

>>> user['last_name'] = 'Doh!'
# Only the last name field will be sent to DynamoDB.
>>> user.partial_save()

```

prepare_full()

Runs through all fields & encodes them to be handed off to DynamoDB as part of an `save (put_item)` call.

Largely internal.

prepare_partial()

Runs through **ONLY** the changed/deleted fields & encodes them to be handed off to DynamoDB as part of an `partial_save (update_item)` call.

Largely internal.

save (overwrite=False)

Saves all data to DynamoDB.

By default, this attempts to ensure that none of the underlying data has changed. If any fields have changed in between when the `Item` was constructed & when it is saved, this call will fail so as not to cause any data loss.

If you're sure possibly overwriting data is acceptable, you can pass an `overwrite=True`. If that's not acceptable, you may be able to use `Item.partial_save` to only write the changed field data.

Optionally accepts an `overwrite` parameter, which should be a boolean. If you provide `True`, the item will be forcibly overwritten within DynamoDB, even if another process changed the data in the meantime. (Default: `False`)

Returns `True` on success, `False` if no save was performed.

Example:

```
>>> user['last_name'] = 'Doh!'
# All data on the Item is sent to DynamoDB.
>>> user.save()

# If it fails, you can overwrite.
>>> user.save(overwrite=True)
```

values()

class boto.dynamodb2.items.**NEWVALUE**

boto.dynamodb2.results

class boto.dynamodb2.results.**BatchGetResultSet** (*args, **kwargs)

fetch_more()

When the iterator runs out of results, this method is run to re-execute the callable (& arguments) to fetch the next page.

Largely internal.

class boto.dynamodb2.results.**ResultSet** (max_page_size=None)

A class used to lazily handle page-to-page navigation through a set of results.

It presents a transparent iterator interface, so that all the user has to do is use it in a typical `for` loop (or list comprehension, etc.) to fetch results, even if they weren't present in the current page of results.

This is used by the `Table.query` & `Table.scan` methods.

Example:

```
>>> users = Table('users')
>>> results = ResultSet()
>>> results.to_call(users.query, username__gte='johndoe')
# Now iterate. When it runs out of results, it'll fetch the next page.
>>> for res in results:
...     print res['username']
```

fetch_more()

When the iterator runs out of results, this method is run to re-execute the callable (& arguments) to fetch the next page.

Largely internal.

first_key

`next ()`

`to_call (the_callable, *args, **kwargs)`

Sets up the callable & any arguments to run it with.

This is stored for subsequent calls so that those queries can be run without requiring user intervention.

Example:

```
# Just an example callable.
>>> def squares_to(y):
...     for x in range(1, y):
...         yield x**2
>>> rs = ResultSet()
# Set up what to call & arguments.
>>> rs.to_call(squares_to, y=3)
```

boto.dynamodb2.table

class `boto.dynamodb2.table.BatchTable (table)`

Used by `Table` as the context manager for batch writes.

You likely don't want to try to use this object directly.

`delete_item (**kwargs)`

`flush ()`

`handle_unprocessed (resp)`

`put_item (data, overwrite=False)`

`resend_unprocessed ()`

`should_flush ()`

class `boto.dynamodb2.table.Table (table_name, schema=None, throughput=None, indexes=None, global_indexes=None, connection=None)`

Interacts & models the behavior of a DynamoDB table.

The `Table` object represents a set (or rough categorization) of records within DynamoDB. The important part is that all records within the table, while largely-schema-free, share the same schema & are essentially namespaced for use in your application. For example, you might have a `users` table or a `forums` table.

Sets up a new in-memory `Table`.

This is useful if the table already exists within DynamoDB & you simply want to use it for additional interactions. The only required parameter is the `table_name`. However, under the hood, the object will call `describe_table` to determine the schema/indexes/throughput. You can avoid this extra call by passing in `schema` & `indexes`.

IMPORTANT - If you're creating a new `Table` for the first time, you should use the `Table.create` method instead, as it will persist the table structure to DynamoDB.

Requires a `table_name` parameter, which should be a simple string of the name of the table.

Optionally accepts a `schema` parameter, which should be a list of `BaseSchemaField` subclasses representing the desired schema.

Optionally accepts a `throughput` parameter, which should be a dictionary. If provided, it should specify a `read` & `write` key, both of which should have an integer value associated with them.

Optionally accepts a `indexes` parameter, which should be a list of `BaseIndexField` subclasses representing the desired indexes.

Optionally accepts a `global_indexes` parameter, which should be a list of `GlobalBaseIndexField` subclasses representing the desired indexes.

Optionally accepts a `connection` parameter, which should be a `DynamoDBConnection` instance (or subclass). This is primarily useful for specifying alternate connection parameters.

Example:

```
# The simple, it-already-exists case.
>>> conn = Table('users')

# The full, minimum-extra-calls case.
>>> from boto import dynamodb2
>>> users = Table('users', schema=[
...     HashKey('username'),
...     RangeKey('date_joined', data_type=NUMBER)
... ], throughput={
...     'read':20,
...     'write': 10,
... }, indexes=[
...     KeysOnlyIndex('MostRecentlyJoined', parts=[
...         HashKey('username')
...         RangeKey('date_joined')
...     ]),
... ], global_indexes=[
...     GlobalAllIndex('UsersByZipcode', parts=[
...         HashKey('zipcode'),
...         RangeKey('username'),
...     ],
...     throughput={
...         'read':10,
...         'write':10,
...     })),
... ], connection=dynamodb2.connect_to_region('us-west-2',
...     aws_access_key_id='key',
...     aws_secret_access_key='key',
... ))
```

batch_get (*keys, consistent=False, attributes=None*)

Fetches many specific items in batch from a table.

Requires a `keys` parameter, which should be a list of dictionaries. Each dictionary should consist of the keys values to specify.

Optionally accepts a `consistent` parameter, which should be a boolean. If you provide `True`, a strongly consistent read will be used. (Default: `False`)

Optionally accepts an `attributes` parameter, which should be a tuple. If you provide any attributes only these will be fetched from DynamoDB.

Returns a `ResultSet`, which transparently handles the pagination of results you get back.

Example:

```
>>> results = users.batch_get(keys=[
...     {
...         'username': 'johndoe',
```

(continues on next page)

(continued from previous page)

```

...     },
...     {
...         'username': 'jane',
...     },
...     {
...         'username': 'fred',
...     },
... ] )
>>> for res in results:
...     print res['first_name']
'John'
'Jane'
'Fred'

```

batch_write()

Allows the batching of writes to DynamoDB.

Since each write/delete call to DynamoDB has a cost associated with it, when loading lots of data, it makes sense to batch them, creating as few calls as possible.

This returns a context manager that will transparently handle creating these batches. The object you get back lightly-resembles a Table object, sharing just the `put_item` & `delete_item` methods (which are all that DynamoDB can batch in terms of writing data).

DynamoDB's maximum batch size is 25 items per request. If you attempt to put/delete more than that, the context manager will batch as many as it can up to that number, then flush them to DynamoDB & continue batching as more calls come in.

Example:

```

# Assuming a table with one record...
>>> with users.batch_write() as batch:
...     batch.put_item(data={
...         'username': 'johndoe',
...         'first_name': 'John',
...         'last_name': 'Doe',
...         'owner': 1,
...     })
...     # Nothing across the wire yet.
...     batch.delete_item(username='bob')
...     # Still no requests sent.
...     batch.put_item(data={
...         'username': 'jane',
...         'first_name': 'Jane',
...         'last_name': 'Doe',
...         'date_joined': 127436192,
...     })
...     # Nothing yet, but once we leave the context, the
...     # put/deletes will be sent.

```

count()

Returns a (very) eventually consistent count of the number of items in a table.

Lag time is about 6 hours, so don't expect a high degree of accuracy.

Example:

```
>>> users.count()
6
```

classmethod create (*table_name*, *schema*, *throughput=None*, *indexes=None*, *global_indexes=None*, *connection=None*)

Creates a new table in DynamoDB & returns an in-memory Table object.

This will setup a brand new table within DynamoDB. The *table_name* must be unique for your AWS account. The *schema* is also required to define the key structure of the table.

IMPORTANT - You should consider the usage pattern of your table up-front, as the schema can **NOT** be modified once the table is created, requiring the creation of a new table & migrating the data should you wish to revise it.

IMPORTANT - If the table already exists in DynamoDB, additional calls to this method will result in an error. If you just need a Table object to interact with the existing table, you should just initialize a new Table object, which requires only the *table_name*.

Requires a *table_name* parameter, which should be a simple string of the name of the table.

Requires a *schema* parameter, which should be a list of BaseSchemaField subclasses representing the desired schema.

Optionally accepts a *throughput* parameter, which should be a dictionary. If provided, it should specify a *read* & *write* key, both of which should have an integer value associated with them.

Optionally accepts a *indexes* parameter, which should be a list of BaseIndexField subclasses representing the desired indexes.

Optionally accepts a *global_indexes* parameter, which should be a list of GlobalBaseIndexField subclasses representing the desired indexes.

Optionally accepts a *connection* parameter, which should be a DynamoDBConnection instance (or subclass). This is primarily useful for specifying alternate connection parameters.

Example:

```
>>> users = Table.create('users', schema=[
...     HashKey('username'),
...     RangeKey('date_joined', data_type=NUMBER)
... ], throughput={
...     'read':20,
...     'write': 10,
... }, indexes=[
...     KeysOnlyIndex('MostRecentlyJoined', parts=[
...         HashKey('username'),
...         RangeKey('date_joined'),
...     ]), global_indexes=[
...         GlobalAllIndex('UsersByZipcode', parts=[
...             HashKey('zipcode'),
...             RangeKey('username'),
...         ]),
...     ],
...     throughput={
...         'read':10,
...         'write':10,
...     }
... ])
```

create_global_secondary_index (*global_index*)

Creates a global index in DynamoDB after the table has been created.

Requires a `global_indexes` parameter, which should be a `GlobalBaseIndexField` subclass representing the desired index.

To update `global_indexes` information on the `Table`, you'll need to call `Table.describe`.

Returns `True` on success.

Example:

```
# To create a global index
>>> users.create_global_secondary_index(
...     global_index=GlobalAllIndex(
...         'TheIndexNameHere', parts=[
...             HashKey('requiredHashkey', data_type=STRING),
...             RangeKey('optionalRangeKey', data_type=STRING)
...         ],
...         throughput={
...             'read': 2,
...             'write': 1,
...         })
... )
True
```

`delete()`

Deletes a table in DynamoDB.

IMPORTANT - Be careful when using this method, there is no undo.

Returns `True` on success.

Example:

```
>>> users.delete()
True
```

`delete_global_secondary_index(global_index_name)`

Deletes a global index in DynamoDB after the table has been created.

Requires a `global_index_name` parameter, which should be a simple string of the name of the global secondary index.

To update `global_indexes` information on the `Table`, you'll need to call `Table.describe`.

Returns `True` on success.

Example:

```
# To delete a global index
>>> users.delete_global_secondary_index('TheIndexNameHere')
True
```

`delete_item(expected=None, conditional_operator=None, **kwargs)`

Deletes a single item. You can perform a conditional delete operation that deletes the item if it exists, or if it has an expected attribute value.

Conditional deletes are useful for only deleting items if specific conditions are met. If those conditions are met, DynamoDB performs the delete. Otherwise, the item is not deleted.

To specify the expected attribute values of the item, you can pass a dictionary of conditions to `expected`. Each condition should follow the pattern `<attributename>__<comparison_operator>=<value_to_expect>`.

IMPORTANT - Be careful when using this method, there is no undo.

To specify the key of the item you'd like to get, you can specify the key attributes as kwargs.

Optionally accepts an `expected` parameter which is a dictionary of expected attribute value conditions.

Optionally accepts a `conditional_operator` which applies to the expected attribute value conditions:

- *AND* - If all of the conditions evaluate to true (default)
- *OR* - True if at least one condition evaluates to true

Returns `True` on success, `False` on failed conditional delete.

Example:

```
# A simple hash key.
>>> users.delete_item(username='johndoe')
True

# A complex hash+range key.
>>> users.delete_item(username='jane', last_name='Doe')
True

# With a key that is an invalid variable name in Python.
# Also, assumes a different schema than previous examples.
>>> users.delete_item(**{
...     'date-joined': 127549192,
... })
True

# Conditional delete
>>> users.delete_item(username='johndoe',
...                    expected={'balance__eq': 0})
True
```

describe()

Describes the current structure of the table in DynamoDB.

This information will be used to update the `schema`, `indexes`, `global_indexes` and `throughput` information on the `Table`. Some calls, such as those involving creating keys or querying, will require this information to be populated.

It also returns the full raw data structure from DynamoDB, in the event you'd like to parse out additional information (such as the `ItemCount` or usage information).

Example:

```
>>> users.describe()
{
    # Lots of keys here...
}
>>> len(users.schema)
2
```

get_item (*consistent=False*, *attributes=None*, ***kwargs*)

Fetches an item (record) from a table in DynamoDB.

To specify the key of the item you'd like to get, you can specify the key attributes as kwargs.

Optionally accepts a `consistent` parameter, which should be a boolean. If you provide `True`, it will perform a consistent (but more expensive) read from DynamoDB. (Default: `False`)

Optionally accepts an `attributes` parameter, which should be a list of fieldname to fetch. (Default: `None`, which means all fields should be fetched)

Returns an `Item` instance containing all the data for that record.

Raises an `ItemNotFound` exception if the item is not found.

Example:

```
# A simple hash key.
>>> john = users.get_item(username='johndoe')
>>> john['first_name']
'John'

# A complex hash+range key.
>>> john = users.get_item(username='johndoe', last_name='Doe')
>>> john['first_name']
'John'

# A consistent read (assuming the data might have just changed).
>>> john = users.get_item(username='johndoe', consistent=True)
>>> john['first_name']
'Johann'

# With a key that is an invalid variable name in Python.
# Also, assumes a different schema than previous examples.
>>> john = users.get_item(**{
...     'date-joined': 127549192,
... })
>>> john['first_name']
'John'
```

`get_key_fields()`

Returns the fields necessary to make a key for a table.

If the `Table` does not already have a populated schema, this will request it via a `Table.describe` call.

Returns a list of fieldnames (strings).

Example:

```
# A simple hash key.
>>> users.get_key_fields()
['username']

# A complex hash+range key.
>>> users.get_key_fields()
['username', 'last_name']
```

`has_item(**kwargs)`

Return whether an item (record) exists within a table in DynamoDB.

To specify the key of the item you'd like to get, you can specify the key attributes as `kwargs`.

Optionally accepts a `consistent` parameter, which should be a boolean. If you provide `True`, it will perform a consistent (but more expensive) read from DynamoDB. (Default: `False`)

Optionally accepts an `attributes` parameter, which should be a list of fieldnames to fetch. (Default: `None`, which means all fields should be fetched)

Returns `True` if an `Item` is present, `False` if not.

Example:

```
# Simple, just hash-key schema.
>>> users.has_item(username='johndoe')
True

# Complex schema, item not present.
>>> users.has_item(
...     username='johndoe',
...     date_joined='2014-01-07'
... )
False
```

lookup (*args, **kwargs)

Look up an entry in DynamoDB. This is mostly backwards compatible with boto.dynamodb. Unlike `get_item`, it takes `hash_key` and `range_key` first, although you may still specify keyword arguments instead.

Also unlike the `get_item` command, if the returned item has no keys (i.e., it does not exist in DynamoDB), a `None` result is returned, instead of an empty key object.

Example::

```
>>> user = users.lookup(username)
>>> user = users.lookup(username, consistent=True)
>>> app = apps.lookup('my_customer_id', 'my_app_id')
```

max_batch_get = 100**new_item** (*args)

Returns a new, blank item

This is mostly for consistency with boto.dynamodb

put_item (data, overwrite=False)

Saves an entire item to DynamoDB.

By default, if any part of the `Item`'s original data doesn't match what's currently in DynamoDB, this request will fail. This prevents other processes from updating the data in between when you read the item & when your request to update the item's data is processed, which would typically result in some data loss.

Requires a `data` parameter, which should be a dictionary of the data you'd like to store in DynamoDB.

Optionally accepts an `overwrite` parameter, which should be a boolean. If you provide `True`, this will tell DynamoDB to blindly overwrite whatever data is present, if any.

Returns `True` on success.

Example:

```
>>> users.put_item(data={
...     'username': 'jane',
...     'first_name': 'Jane',
...     'last_name': 'Doe',
...     'date_joined': 126478915,
... })
True
```

query (limit=None, index=None, reverse=False, consistent=False, attributes=None, max_page_size=None, **filter_kwargs)

WARNING: This method is provided **strictly** for backward-compatibility. It returns results in an incorrect order.

If you are writing new code, please use `Table.query_2`.

query_2 (*limit=None, index=None, reverse=False, consistent=False, attributes=None, max_page_size=None, query_filter=None, conditional_operator=None, **filter_kwargs*)
 Queries for a set of matching items in a DynamoDB table.

Queries can be performed against a hash key, a hash+range key or against any data stored in your local secondary indexes. Query filters can be used to filter on arbitrary fields.

Note - You can not query against arbitrary fields within the data stored in DynamoDB unless you specify `query_filter` values.

To specify the filters of the items you'd like to get, you can specify the filters as kwargs. Each filter kwarg should follow the pattern `<fieldname>__<filter_operation>=<value_to_look_for>`. Query filters are specified in the same way.

Optionally accepts a `limit` parameter, which should be an integer count of the total number of items to return. (Default: `None` - all results)

Optionally accepts an `index` parameter, which should be a string of name of the local secondary index you want to query against. (Default: `None`)

Optionally accepts a `reverse` parameter, which will present the results in reverse order. (Default: `False` - normal order)

Optionally accepts a `consistent` parameter, which should be a boolean. If you provide `True`, it will force a consistent read of the data (more expensive). (Default: `False` - use eventually consistent reads)

Optionally accepts a `attributes` parameter, which should be a tuple. If you provide any attributes only these will be fetched from DynamoDB. This uses the `AttributesToGet` and set's `Select to SPECIFIC_ATTRIBUTES` API.

Optionally accepts a `max_page_size` parameter, which should be an integer count of the maximum number of items to retrieve **per-request**. This is useful in making faster requests & prevent the scan from drowning out other queries. (Default: `None` - fetch as many as DynamoDB will return)

Optionally accepts a `query_filter` which is a dictionary of filter conditions against any arbitrary field in the returned data.

Optionally accepts a `conditional_operator` which applies to the query filter conditions:

- *AND* - True if all filter conditions evaluate to true (default)
- *OR* - True if at least one filter condition evaluates to true

Returns a `ResultSet` containing `"Item"`s, which transparently handles the pagination of results you get back.

Example:

```
# Look for last names equal to "Doe".
>>> results = users.query(last_name__eq='Doe')
>>> for res in results:
...     print res['first_name']
'John'
'Jane'

# Look for last names beginning with "D", in reverse order, limit 3.
>>> results = users.query(
...     last_name__startswith='D',
...     reverse=True,
...     limit=3
... )
>>> for res in results:
```

(continues on next page)

(continued from previous page)

```

...     print res['first_name']
'Alice'
'Jane'
'John'

# Use an LSI & a consistent read.
>>> results = users.query(
...     date_joined__gte=1236451000,
...     owner__eq=1,
...     index='DateJoinedIndex',
...     consistent=True
... )
>>> for res in results:
...     print res['first_name']
'Alice'
'Bob'
'John'
'Fred'

# Filter by non-indexed field(s)
>>> results = users.query(
...     last_name__eq='Doe',
...     reverse=True,
...     query_filter={
...         'first_name__startswith': 'A'
...     }
... )
>>> for res in results:
...     print res['first_name'] + ' ' + res['last_name']
'Alice Doe'

```

query_count (*index=None, consistent=False, conditional_operator=None, query_filter=None, scan_index_forward=True, limit=None, exclusive_start_key=None, **filter_kwargs*)
 Queries the exact count of matching items in a DynamoDB table.

Queries can be performed against a hash key, a hash+range key or against any data stored in your local secondary indexes. Query filters can be used to filter on arbitrary fields.

To specify the filters of the items you'd like to get, you can specify the filters as kwargs. Each filter kwarg should follow the pattern `<fieldname>__<filter_operation>=<value_to_look_for>`. Query filters are specified in the same way.

Optionally accepts an `index` parameter, which should be a string of name of the local secondary index you want to query against. (Default: None)

Optionally accepts a `consistent` parameter, which should be a boolean. If you provide `True`, it will force a consistent read of the data (more expensive). (Default: `False` - use eventually consistent reads)

Optionally accepts a `query_filter` which is a dictionary of filter conditions against any arbitrary field in the returned data.

Optionally accepts a `conditional_operator` which applies to the query filter conditions:

- *AND* - True if all filter conditions evaluate to true (default)
- *OR* - True if at least one filter condition evaluates to true

Optionally accept a `exclusive_start_key` which is used to get the remaining items when a query cannot return the complete count.

Returns an integer which represents the exact amount of matched items.

Parameters `scan_index_forward` (*boolean*) – Specifies ascending (true) or descending (false) traversal of the index. DynamoDB returns results reflecting the requested order determined by the range key. If the data type is Number, the results are returned in numeric order. For String, the results are returned in order of ASCII character code values. For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

If ScanIndexForward is not specified, the results are returned in ascending order.

Parameters `limit` (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items).

Example:

```
# Look for last names equal to "Doe".
>>> users.query_count(last_name__eq='Doe')
5

# Use an LSI & a consistent read.
>>> users.query_count(
...     date_joined__gte=1236451000,
...     owner__eq=1,
...     index='DateJoinedIndex',
...     consistent=True
... )
2
```

scan (*limit=None, segment=None, total_segments=None, max_page_size=None, attributes=None, conditional_operator=None, **filter_kwargs*)

Scans across all items within a DynamoDB table.

Scans can be performed against a hash key or a hash+range key. You can additionally filter the results after the table has been read but before the response is returned by using query filters.

To specify the filters of the items you'd like to get, you can specify the filters as kwargs. Each filter kwarg should follow the pattern `<fieldname>__<filter_operation>=<value_to_look_for>`.

Optionally accepts a `limit` parameter, which should be an integer count of the total number of items to return. (Default: `None` - all results)

Optionally accepts a `segment` parameter, which should be an integer of the segment to retrieve on. Please see the documentation about Parallel Scans (Default: `None` - no segments)

Optionally accepts a `total_segments` parameter, which should be an integer count of number of segments to divide the table into. Please see the documentation about Parallel Scans (Default: `None` - no segments)

Optionally accepts a `max_page_size` parameter, which should be an integer count of the maximum number of items to retrieve **per-request**. This is useful in making faster requests & prevent the scan from drowning out other queries. (Default: `None` - fetch as many as DynamoDB will return)

Optionally accepts an `attributes` parameter, which should be a tuple. If you provide any attributes only these will be fetched from DynamoDB. This uses the `AttributesToGet` and set's `Select` to `SPECIFIC_ATTRIBUTES` API.

Returns a `ResultSet`, which transparently handles the pagination of results you get back.

Example:

```
# All results.
>>> everything = users.scan()
```

(continues on next page)

(continued from previous page)

```

# Look for last names beginning with "D".
>>> results = users.scan(last_name__beginswith='D')
>>> for res in results:
...     print res['first_name']
'Alice'
'John'
'Jane'

# Use an ``IN`` filter & limit.
>>> results = users.scan(
...     age__in=[25, 26, 27, 28, 29],
...     limit=1
... )
>>> for res in results:
...     print res['first_name']
'Alice'

```

update (*throughput=None, global_indexes=None*)

Updates table attributes and global indexes in DynamoDB.

Optionally accepts a `throughput` parameter, which should be a dictionary. If provided, it should specify a read & write key, both of which should have an integer value associated with them.

Optionally accepts a `global_indexes` parameter, which should be a dictionary. If provided, it should specify the index name, which is also a dict containing a read & write key, both of which should have an integer value associated with them. If you are writing new code, please use `Table.update_global_secondary_index`.

Returns True on success.

Example:

```

# For a read-heavier application...
>>> users.update(throughput={
...     'read': 20,
...     'write': 10,
... })
True

# To also update the global index(es) throughput.
>>> users.update(throughput={
...     'read': 20,
...     'write': 10,
... },
... global_secondary_indexes={
...     'TheIndexNameHere': {
...         'read': 15,
...         'write': 5,
...     }
... })
True

```

update_global_secondary_index (*global_indexes*)

Updates a global index(es) in DynamoDB after the table has been created.

Requires a `global_indexes` parameter, which should be a dictionary. If provided, it should specify the index name, which is also a dict containing a read & write key, both of which should have an

integer value associated with them.

To update `global_indexes` information on the `Table`, you'll need to call `Table.describe`.

Returns `True` on success.

Example:

```
# To update a global index
>>> users.update_global_secondary_index(global_indexes={
...     'TheIndexNameHere': {
...         'read': 15,
...         'write': 5,
...     }
... })
True
```

`use_boolean()`

4.125.2 Low-Level API

`boto.dynamodb2`

`boto.dynamodb2.connect_to_region(region_name, **kw_params)`

`boto.dynamodb2.regions()`

Get all available regions for the Amazon DynamoDB service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

`boto.dynamodb2.layer1`

class `boto.dynamodb2.layer1.DynamoDBConnection(**kwargs)`

Amazon DynamoDB **Overview**

This is the Amazon DynamoDB API Reference. This guide provides descriptions and samples of the low-level DynamoDB API. For information about DynamoDB application development, go to the [‘Amazon DynamoDB Developer Guide’](#).

Instead of making the requests to the low-level DynamoDB API directly from your application, we recommend that you use the AWS Software Development Kits (SDKs). The easy-to-use libraries in the AWS SDKs make it unnecessary to call the low-level DynamoDB API directly from your application. The libraries take care of request authentication, serialization, and connection management. For more information, go to [‘Using the AWS SDKs with DynamoDB’](#) in the Amazon DynamoDB Developer Guide .

If you decide to code against the low-level DynamoDB API directly, you will need to write the necessary code to authenticate your requests. For more information on signing your requests, go to [‘Using the DynamoDB API’](#) in the Amazon DynamoDB Developer Guide .

The following are short descriptions of each low-level API action, organized by function.

Managing Tables

- `CreateTable` - Creates a table with user-specified provisioned throughput settings. You must designate one attribute as the hash primary key for the table; you can optionally designate a second attribute as the range primary key. DynamoDB creates indexes on these key attributes for fast data access. Optionally, you can create one or more secondary indexes, which provide fast data access using non-key attributes.
- `DescribeTable` - Returns metadata for a table, such as table size, status, and index information.

- UpdateTable - Modifies the provisioned throughput settings for a table. Optionally, you can modify the provisioned throughput settings for global secondary indexes on the table.
- ListTables - Returns a list of all tables associated with the current AWS account and endpoint.
- DeleteTable - Deletes a table and all of its indexes.

For conceptual information about managing tables, go to [‘Working with Tables’](#) in the Amazon DynamoDB Developer Guide .

Reading Data

- GetItem - Returns a set of attributes for the item that has a given primary key. By default, GetItem performs an eventually consistent read; however, applications can specify a strongly consistent read instead.
- BatchGetItem - Performs multiple GetItem requests for data items using their primary keys, from one table or multiple tables. The response from BatchGetItem has a size limit of 16 MB and returns a maximum of 100 items. Both eventually consistent and strongly consistent reads can be used.
- Query - Returns one or more items from a table or a secondary index. You must provide a specific hash key value. You can narrow the scope of the query using comparison operators against a range key value, or on the index key. Query supports either eventual or strong consistency. A single response has a size limit of 1 MB.
- Scan - Reads every item in a table; the result set is eventually consistent. You can limit the number of items returned by filtering the data attributes, using conditional expressions. Scan can be used to enable ad-hoc querying of a table against non-key attributes; however, since this is a full table scan without using an index, Scan should not be used for any application query use case that requires predictable performance.

For conceptual information about reading data, go to [‘Working with Items’](#) and [‘Query and Scan Operations’](#) in the Amazon DynamoDB Developer Guide .

Modifying Data

- PutItem - Creates a new item, or replaces an existing item with a new item (including all the attributes). By default, if an item in the table already exists with the same primary key, the new item completely replaces the existing item. You can use conditional operators to replace an item only if its attribute values match certain conditions, or to insert a new item only if that item doesn’t already exist.
- UpdateItem - Modifies the attributes of an existing item. You can also use conditional operators to perform an update only if the item’s attribute values match certain conditions.
- DeleteItem - Deletes an item in a table by primary key. You can use conditional operators to perform a delete an item only if the item’s attribute values match certain conditions.
- BatchWriteItem - Performs multiple PutItem and DeleteItem requests across multiple tables in a single request. A failure of any request(s) in the batch will not cause the entire BatchWriteItem operation to fail. Supports batches of up to 25 items to put or delete, with a maximum total request size of 16 MB.

For conceptual information about modifying data, go to [‘Working with Items’](#) and [‘Query and Scan Operations’](#) in the Amazon DynamoDB Developer Guide .

```
APIVersion = '2012-08-10'
```

```
DefaultRegionEndpoint = 'dynamodb.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
NumberRetries = 10
```

```
ResponseError
```

```
alias of boto.exception.JSONResponseError
```

```
ServiceName = 'DynamoDB'
```

```
TargetPrefix = 'DynamoDB_20120810'
```

```
batch_get_item (request_items, return_consumed_capacity=None)
```

The BatchGetItem operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

A single operation can retrieve up to 16 MB of data, which can contain as many as 100 items.

BatchGetItem will return a partial result if the response size limit is exceeded, the table's provisioned throughput is exceeded, or an internal processing failure occurs. If a partial result is returned, the operation returns a value for UnprocessedKeys . You can use this value to retry the operation starting with the next item to get.

For example, if you ask to retrieve 100 items, but each individual item is 300 KB in size, the system returns 52 items (so as not to exceed the 16 MB limit). It also returns an appropriate UnprocessedKeys value so you can get the next page of results. If desired, your application can include its own logic to assemble the pages of results into one data set.

If none of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then BatchGetItem will return a ProvisionedThroughputExceededException . If at least one of the items is successfully processed, then BatchGetItem completes successfully, while returning the keys of the unread items in UnprocessedKeys .

If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, we strongly recommend that you use an exponential backoff algorithm . If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, go to [‘Batch Operations and Error Handling’](#) in the Amazon DynamoDB Developer Guide .

By default, BatchGetItem performs eventually consistent reads on every table in the request. If you want strongly consistent reads instead, you can set ConsistentRead to *True* for any or all tables.

In order to minimize response latency, BatchGetItem retrieves items in parallel.

When designing your application, keep in mind that DynamoDB does not return attributes in any particular order. To help parse the response by item, include the primary key values for the items in your request in the AttributesToGet parameter.

If a requested item does not exist, it is not returned in the result. Requests for nonexistent items consume the minimum read capacity units according to the type of read. For more information, see [‘Capacity Units Calculations’](#) in the Amazon DynamoDB Developer Guide .

Parameters `request_items` (*map*) –

A map of one or more table names and, for each table, the corresponding primary keys for the items to retrieve. Each table name can be invoked only once.

Each element in the map consists of the following:

- **Keys** - **An array of primary key attribute values that define specific** items in the table. For each primary key, you must provide all of the key attributes. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.
- **AttributesToGet** - **One or more attributes to be retrieved from the** table. By default, all attributes are returned. If a specified attribute is not found, it does not appear in the result. Note that AttributesToGet has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.
- **ConsistentRead** - **If *True*, a strongly consistent read is used; if *False* (the default), an eventually consistent read is used.**

Parameters `return_consumed_capacity` (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.

`batch_write_item` (*request_items*, *return_consumed_capacity=None*, *return_item_collection_metrics=None*)

The BatchWriteItem operation puts or deletes multiple items in one or more tables. A single call to BatchWriteItem can write up to 16 MB of data, which can comprise as many as 25 put or delete requests. Individual items to be written can be as large as 400 KB.

BatchWriteItem cannot update items. To update items, use the UpdateItem API.

The individual PutItem and DeleteItem operations specified in BatchWriteItem are atomic; however BatchWriteItem as a whole is not. If any requested operations fail because the table's provisioned throughput is exceeded or an internal processing failure occurs, the failed operations are returned in the UnprocessedItems response parameter. You can investigate and optionally resend the requests. Typically, you would call BatchWriteItem in a loop. Each iteration would check for unprocessed items and submit a new BatchWriteItem request with those unprocessed items until all items have been processed.

Note that if none of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then BatchWriteItem will return a ProvisionedThroughputExceededException .

If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, we strongly recommend that you use an exponential backoff algorithm . If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, go to **'Batch Operations and Error Handling'** in the Amazon DynamoDB Developer Guide .

With BatchWriteItem , you can efficiently write or delete large amounts of data, such as from Amazon Elastic MapReduce (EMR), or copy data from another database into DynamoDB. In order to improve performance with these large-scale operations, BatchWriteItem does not behave in the same way as individual PutItem and DeleteItem calls would. For example, you cannot specify conditions on individual put and delete requests, and BatchWriteItem does not return deleted items in the response.

If you use a programming language that supports concurrency, such as Java, you can use threads to write items in parallel. Your application must include the necessary logic to manage the threads. With languages that don't support threading, such as PHP, you must update or delete the specified items one at a time. In both situations, BatchWriteItem provides an alternative where the API performs the specified put and delete operations in parallel, giving you the power of the thread pool approach without having to introduce complexity into your application.

Parallel processing reduces latency, but each specified put and delete request consumes the same number of write capacity units whether it is processed in parallel or not. Delete operations on nonexistent items consume one write capacity unit.

If one or more of the following is true, DynamoDB rejects the entire batch write operation:

- One or more tables specified in the BatchWriteItem request does not exist.
- Primary key attributes specified on an item in the request do not match those in the corresponding table's primary key schema.
- You try to perform multiple operations on the same item in the same BatchWriteItem request. For example, you cannot put and delete the same item in the same BatchWriteItem request.
- There are more than 25 requests in the batch.
- Any individual item in a batch exceeds 400 KB.
- The total request size exceeds 16 MB.

Parameters `request_items` (*map*) –

A **map of one or more table names and, for each table, a list of** operations to be performed (DeleteRequest or PutRequest). Each element in the map consists of the following:

- **DeleteRequest - Perform a DeleteItem operation on the specified item.**

The item to be deleted is identified by a Key subelement:

- **Key** - A map of primary key attribute values that uniquely identify the item. Each entry in this map consists of an attribute name and an attribute value. For each primary key, you must provide all of the key attributes. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.

- **PutRequest** - Perform a PutItem operation on the specified item. The

item to be put is identified by an Item subelement:

- **Item** - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a ValidationException exception. If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

Parameters

- **return_consumed_capacity** (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.
- **return_item_collection_metrics** (*string*) – A value that if set to *SIZE*, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.

create_table (*attribute_definitions*, *table_name*, *key_schema*, *provisioned_throughput*, *local_secondary_indexes=None*, *global_secondary_indexes=None*)

The CreateTable operation adds a new table to your account. In an AWS account, table names must be unique within each region. That is, you can have two tables with same name if you create the tables in different regions.

CreateTable is an asynchronous operation. Upon receiving a CreateTable request, DynamoDB immediately returns a response with a TableStatus of *CREATING*. After the table is created, DynamoDB sets the TableStatus to *ACTIVE*. You can perform read and write operations only on an *ACTIVE* table.

You can optionally define secondary indexes on the new table, as part of the CreateTable operation. If you want to create multiple tables with secondary indexes on them, you must create the tables sequentially. Only one table with secondary indexes can be in the *CREATING* state at any given time.

You can use the DescribeTable API to check the table status.

Parameters

- **attribute_definitions** (*list*) – An array of attributes that describe the key schema for the table and indexes.
- **table_name** (*string*) – The name of the table to create.
- **key_schema** (*list*) – Specifies the attributes that make up the primary key for a table or an index. The attributes in KeySchema must also be defined in the AttributeDefinitions array. For more information, see **'Data Model'** in the Amazon DynamoDB Developer Guide .

Each KeySchemaElement in the array is composed of:

- `AttributeName` - The name of this key attribute.
- `KeyType` - Determines whether the key attribute is *HASH* or *RANGE*.

For a primary key that consists of a hash attribute, you must specify exactly one element with a `KeyType` of *HASH*.

For a primary key that consists of hash and range attributes, you must specify exactly two elements, in this order: The first element must have a `KeyType` of *HASH*, and the second element must have a `KeyType` of *RANGE*.

For more information, see ‘Specifying the Primary Key’_ in the Amazon DynamoDB Developer Guide .

Parameters `local_secondary_indexes` (*list*)–

One or more local secondary indexes (the maximum is five) to be created on the table. Each index is scoped to a given hash key value. There is a 10 GB size limit per hash key; otherwise, the size of a local secondary index is unconstrained.

Each local secondary index in the array includes the following:

- **IndexName** - **The name of the local secondary index. Must be unique** only for this table.
- **KeySchema** - **Specifies the key schema for the local secondary index.** The key schema must begin with the same hash key attribute as the table.
- **Projection** - **Specifies attributes that are copied (projected) from**

the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:

– `ProjectionType` - One of the following:

- * ***KEYS_ONLY*** - **Only the index and primary keys are projected into the** index.
- * ***INCLUDE*** - **Only the specified table attributes are projected into the** index. The list of projected attributes are in `NonKeyAttributes` .
- * ***ALL*** - All of the table attributes are projected into the index.

– **`NonKeyAttributes`** - **A list of one or more non-key attribute names that** are projected into the secondary index. The total count of attributes specified in `NonKeyAttributes` , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

Parameters `global_secondary_indexes` (*list*)–

One or more global secondary indexes (the maximum is five) to be created on the table. Each global secondary index in the array includes the following:

- **IndexName** - **The name of the global secondary index. Must be unique** only for this table.
- **KeySchema** - Specifies the key schema for the global secondary index.
- **Projection** - **Specifies attributes that are copied (projected) from**

the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:

– `ProjectionType` - One of the following:

- * ***KEYS_ONLY*** - **Only the index and primary keys are projected into the** index.

* **INCLUDE** - Only the specified table attributes are projected into the index. The list of projected attributes are in `NonKeyAttributes` .

* **ALL** - All of the table attributes are projected into the index.

– **NonKeyAttributes** - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes specified in `NonKeyAttributes` , summed across all of the secondary indexes, must not exceed 20. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total.

• **ProvisionedThroughput** - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units.

Parameters `provisioned_throughput` (*dict*) – Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the `UpdateTable` operation.

For current minimum and maximum provisioned throughput values, see ‘Limits’_ in the Amazon DynamoDB Developer Guide .

delete_item (*table_name*, *key*, *expected=None*, *conditional_operator=None*, *return_values=None*, *return_consumed_capacity=None*, *return_item_collection_metrics=None*, *condition_expression=None*, *expression_attribute_names=None*, *expression_attribute_values=None*)

Deletes a single item in a table by primary key. You can perform a conditional delete operation that deletes the item if it exists, or if it has an expected attribute value.

In addition to deleting an item, you can also return the item’s attribute values in the same operation, using the `ReturnValues` parameter.

Unless you specify conditions, the `DeleteItem` is an idempotent operation; running it multiple times on the same item or attribute does not result in an error response.

Conditional deletes are useful for deleting items only if specific conditions are met. If those conditions are met, DynamoDB performs the delete. Otherwise, the item is not deleted.

Parameters

- **table_name** (*string*) – The name of the table from which to delete the item.
- **key** (*map*) – A map of attribute names to `AttributeValue` objects, representing the primary key of the item to delete.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.

Parameters `expected` (*map*) –

There is a newer parameter available. Use `ConditionExpression` instead. Note that if you use `Expected` and `ConditionExpression` at the same time, DynamoDB will return a `ValidationException` exception.

This parameter does not support lists or maps.

A map of attribute/condition pairs. `Expected` provides a conditional block for the `DeleteItem` operation.

Each element of `Expected` consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each `Expected` element, the result of the evaluation is either true or false.

If you specify more than one element in the `Expected` map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the `Conditional-`

Operator parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the Expected map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- **AttributeValueList - One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the ComparisonOperator being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see [‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’](http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters). For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions.
- **ComparisonOperator - A comparator for evaluating attributes in the**

AttributeValueList . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: *EQ* | *NE* | *LE* | *LT* | *GE* | *GT* | *NOT_NULL* | *NULL* | *CONTAINS* | *NOT_CONTAINS* | *BEGINS_WITH* | *IN* | *BETWEEN* The following are descriptions of each comparison operator.

- ***EQ*** [Equal. *EQ* is supported for all datatypes, including lists] and maps. AttributeValueList can contain only one AttributeValue element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >
- ***NE*** [Not equal. *NE* is supported for all datatypes, including] lists and maps. AttributeValueList can contain only one AttributeValue of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an AttributeValue of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >
- ***LE*** [Less than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- ***LT*** [Less than. AttributeValueList can contain only one] AttributeValue of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- ***GE*** [Greater than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- ***GT*** [Greater than. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >

- **NOT_NULL** [The attribute exists. *NOT_NULL* is supported for all] datatypes, including lists and maps. This operator tests for the existence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NOT_NULL*, the result is a Boolean true . This result is because the attribute "a" exists; its data type is not relevant to the *NOT_NULL* comparison operator.
- **NULL** [The attribute does not exist. *NULL* is supported for all] datatypes, including lists and maps. This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NULL*, the result is a Boolean false . This is because the attribute "a" exists; its data type is not relevant to the *NULL* comparison operator.
- **CONTAINS** [Checks for a subsequence, or value in a set.] AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating "a CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- **NOT_CONTAINS** [Checks for absence of a subsequence, or absence of a] value in a set. AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it does not find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- **BEGINS_WITH** [Checks for a prefix. AttributeValueList can contain] only one AttributeValue of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type). >
- **IN** [Checks for matching elements within two sets.] AttributeValueList can contain one or more AttributeValue elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** [Greater than or equal to the first value, and less than] or equal to the second value. AttributeValueList must contain two AttributeValue elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, {"S": "6"} does not compare to {"N": "6"}. Also, {"N": "6"} does not compare to {"NS": ["6", "2", "1"]}

For usage examples of AttributeValueList and ComparisonOperator , see '[Legacy Conditional Parameters](#)' in the Amazon DynamoDB Developer Guide .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of AttributeValueList and ComparisonOperator :

- Value - A value for DynamoDB to compare with an attribute.
- **Exists** - A Boolean value that causes DynamoDB to evaluate the value

before attempting the conditional operation:

- If **Exists** is *True*, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- If **Exists** is *False*, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for **Exists** is *True*.

The Value and Exists parameters are incompatible with `AttributeValueList` and `ComparisonOperator`. Note that if you use both sets of parameters at once, DynamoDB will return a `ValidationException`.

Parameters `conditional_operator` (*string*) -

There is a newer parameter available. Use `ConditionExpression` instead. Note that if you use `ConditionalOperator` and `ConditionExpression` at the same time, DynamoDB will return a `ValidationException`.

This parameter does not support lists or maps.

A logical operator to apply to the conditions in the `Expected` map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit `ConditionalOperator`, then **AND** is the default.

The operation will succeed only if the entire map evaluates to true.

Parameters `return_values` (*string*) -

Use `ReturnValues` if you want to get the item attributes as they appeared before they were deleted. For `DeleteItem`, the valid values are:

- **NONE** - If `ReturnValues` is not specified, or if its value is **NONE**, then nothing is returned. (This setting is the default for `ReturnValues`.)
- **ALL_OLD** - The content of the old item is returned.

Parameters

- **return_consumed_capacity** (*string*) - A value that if set to **TOTAL**, the response includes `ConsumedCapacity` data for tables and indexes. If set to **INDEXES**, the response includes `ConsumedCapacity` for indexes. If set to **NONE** (the default), `ConsumedCapacity` is not included in the response.
- **return_item_collection_metrics** (*string*) - A value that if set to **SIZE**, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to **NONE** (the default), no statistics are returned.
- **condition_expression** (*string*) - A condition that must be satisfied in order for a conditional `DeleteItem` to succeed.

An expression can contain any of the following:

- **Boolean functions:** `'attribute_exists | attribute_not_exists | contains | begins_with'` These function names are case-sensitive.

- **Comparison operators:** ‘ = | <> | < | > | <= >= | BETWEEN | IN ‘
- **Logical operators:** AND | OR | NOT

For more information on condition expressions, go to ‘**Specifying Conditions**’_ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_names` (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using `ExpressionAttributeNames` :

- To shorten an attribute name that is very long or unwieldy in an expression.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following expression:

- ‘`order.customerInfo.LastName = “Smith” OR order.customerInfo.LastName = “Jones”`’

Now suppose that you specified the following for `ExpressionAttributeNames` :

- { “#name”: “order.customerInfo.LastName” }

The expression can now be simplified as follows:

- `#name = “Smith” OR #name = “Jones”`

For more information on expression attribute names, go to ‘**Accessing Item Attributes**’_ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_values` (*map*) – One or more values that can be substituted in an expression.

Use the : (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ “:avail”:{“S”:“Available”}, “:back”:{“S”:“Backordered”}, “:disc”:{“S”:“Discontinued”} }
```

You could then use these values in an expression, such as this:

ProductStatus IN (:avail, :back, :disc)

For more information on expression attribute values, go to ‘**Specifying Conditions**’_ in the Amazon DynamoDB Developer Guide .

delete_table (*table_name*)

The `DeleteTable` operation deletes a table and all of its items. After a `DeleteTable` request, the specified table is in the *DELETING* state until DynamoDB completes the deletion. If the table is in the *ACTIVE* state, you can delete it. If a table is in *CREATING* or *UPDATING* states, then DynamoDB returns a `ResourceInUseException` . If the specified table does not exist, DynamoDB returns a `ResourceNotFoundException` . If table is already in the *DELETING* state, no error is returned.

DynamoDB might continue to accept data read and write operations, such as `GetItem` and `PutItem` , on a table in the *DELETING* state until the table deletion is complete.

When you delete a table, any indexes on that table are also deleted.

Use the `DescribeTable` API to check the status of the table.

Parameters `table_name` (*string*) – The name of the table to delete.

describe_table (*table_name*)

Returns information about the table, including the current status of the table, when it was created, the primary key schema, and any indexes on the table.

If you issue a DescribeTable request immediately after a CreateTable request, DynamoDB might return a ResourceNotFoundException. This is because DescribeTable uses an eventually consistent query, and the metadata for your table might not be available at that moment. Wait for a few seconds, and then try the DescribeTable request again.

Parameters *table_name* (*string*) – The name of the table to describe.

get_item (*table_name*, *key*, *attributes_to_get=None*, *consistent_read=None*, *return_consumed_capacity=None*, *projection_expression=None*, *projection_attribute_names=None*)

The GetItem operation returns a set of attributes for the item with the given primary key. If there is no matching item, GetItem does not return any data.

GetItem provides an eventually consistent read by default. If your application requires a strongly consistent read, set ConsistentRead to *True*. Although a strongly consistent read might take more time than an eventually consistent read, it always returns the last updated value.

Parameters

- **table_name** (*string*) – The name of the table containing the requested item.
- **key** (*map*) – A map of attribute names to AttributeValue objects, representing the primary key of the item to retrieve.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.

Parameters *attributes_to_get* (*list*) –

There is a newer parameter available. Use ProjectionExpression instead. Note that if you use AttributesToGet and ProjectionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter allows you to retrieve lists or maps; however, it cannot retrieve individual list or map elements.

The names of one or more attributes to retrieve. If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that AttributesToGet has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

Parameters

- **consistent_read** (*boolean*) – A value that if set to *True*, then the operation uses strongly consistent reads; otherwise, eventually consistent reads are used.
- **return_consumed_capacity** (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.
- **projection_expression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information on projection expressions, go to ‘Accessing Item Attributes’_ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_names` (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using `ExpressionAttributeNames` :

- To shorten an attribute name that is very long or unwieldy in an expression.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following expression:

- ‘`order.customerInfo.LastName = “Smith” OR order.customerInfo.LastName = “Jones”`’

Now suppose that you specified the following for `ExpressionAttributeNames` :

- {`“#name”:”order.customerInfo.LastName”`}

The expression can now be simplified as follows:

- `#name = “Smith” OR #name = “Jones”`

For more information on expression attribute names, go to ‘Accessing Item Attributes’_ in the Amazon DynamoDB Developer Guide .

list_tables (*exclusive_start_table_name=None, limit=None*)

Returns an array of table names associated with the current account and endpoint. The output from `ListTables` is paginated, with each page returning a maximum of 100 table names.

Parameters

- **exclusive_start_table_name** (*string*) – The first table name that this operation will evaluate. Use the value that was returned for `LastEvaluatedTableName` in a previous operation, so that you can obtain the next page of results.
- **limit** (*integer*) – A maximum number of table names to return. If this parameter is not specified, the limit is 100.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

put_item (*table_name, item, expected=None, return_values=None, return_consumed_capacity=None, return_item_collection_metrics=None, conditional_operator=None, condition_expression=None, expression_attribute_names=None, expression_attribute_values=None*)

Creates a new item, or replaces an old item with a new item. If an item that has the same primary key as the new item already exists in the specified table, the new item completely replaces the existing item. You can perform a conditional put operation (add a new item if one with the specified primary key doesn’t exist), or replace an existing item if it has certain attribute values.

In addition to putting an item, you can also return the item’s attribute values in the same operation, using the `ReturnValues` parameter.

When you add an item, the primary key attribute(s) are the only required attributes. Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes cannot be empty. Requests with empty values will be rejected with a `ValidationException` exception.

You can request that `PutItem` return either a copy of the original item (before the update) or a copy of the updated item (after the update). For more information, see the `ReturnValues` description below.

To prevent a new item from replacing an existing item, use a conditional put operation with ComparisonOperator set to *NULL* for the primary key attribute, or attributes.

For more information about using this API, see ‘**Working with Items**’_ in the Amazon DynamoDB Developer Guide .

Parameters

- **table_name** (*string*) – The name of the table to contain the item.
- **item** (*map*) – A map of attribute name/value pairs, one for each attribute. Only the primary key attributes are required; you can optionally provide other attribute name-value pairs for the item.

You must provide all of the attributes for the primary key. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table’s attribute definition.

For more information about primary keys, see ‘Primary Key’_ in the Amazon DynamoDB Developer Guide .

Each element in the Item map is an AttributeValue object.

Parameters **expected** (*map*) –

There is a newer parameter available. Use ConditionExpression instead. Note that if you use Expected and ConditionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A map of attribute/condition pairs. Expected provides a conditional block for the PutItem operation. **Each element of Expected consists of an attribute name, a comparison** operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each Expected element, the result of the evaluation is either true or false.

If you specify more than one element in the Expected map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the ConditionalOperator parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the Expected map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- **AttributeValueList - One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the ComparisonOperator being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see ‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’_. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions.
- **ComparisonOperator - A comparator for evaluating attributes in the**

AttributeValueList . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: *EQ* | *NE* | *LE* | *LT* | *GE* | *GT* | *NOT_NULL* | *NULL* | *CONTAINS* | *NOT_CONTAINS* | *BEGINS_WITH* | *IN* | *BETWEEN* The following are descriptions of each comparison operator.

- ***EQ*** [Equal. *EQ* is supported for all datatypes, including lists] and maps. AttributeValueList can contain only one AttributeValue element of type String, Number, Binary,

String Set, Number Set, or Binary Set. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >

- **NE** [Not equal. *NE* is supported for all datatypes, including] lists and maps. AttributeValueList can contain only one AttributeValue of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an AttributeValue of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >
- **LE** [Less than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **LT** [Less than. AttributeValueList can contain only one] AttributeValue of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **GE** [Greater than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **GT** [Greater than. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **NOT_NULL** [The attribute exists. *NOT_NULL* is supported for all] datatypes, including lists and maps. This operator tests for the existence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NOT_NULL*, the result is a Boolean true . This result is because the attribute "a" exists; its data type is not relevant to the *NOT_NULL* comparison operator.
- **NULL** [The attribute does not exist. *NULL* is supported for all] datatypes, including lists and maps. This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NULL*, the result is a Boolean false . This is because the attribute "a" exists; its data type is not relevant to the *NULL* comparison operator.
- **CONTAINS** [Checks for a subsequence, or value in a set.] AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating "a CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- **NOT_CONTAINS** [Checks for absence of a subsequence, or absence of a] value in a set. AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it does not find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- **BEGINS_WITH** [Checks for a prefix. AttributeValueList can contain] only one AttributeValue of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type). >
- **IN** [Checks for matching elements within two sets.] AttributeValueList can contain one or more AttributeValue elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** [Greater than or equal to the first value, and less than] or equal to the second value. AttributeValueList must contain two AttributeValue elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, {"S": "6"} does not compare to {"N": "6"}. Also, {"N": "6"} does not compare to {"NS": ["6", "2", "1"]}

For usage examples of AttributeValueList and ComparisonOperator , see '[Legacy Conditional Parameters](#)' in the Amazon DynamoDB Developer Guide .

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of AttributeValueList and ComparisonOperator :

- Value - A value for DynamoDB to compare with an attribute.
- **Exists - A Boolean value that causes DynamoDB to evaluate the value**

before attempting the conditional operation:

- **If Exists is True, DynamoDB will check to see if that attribute** value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- **If Exists is False, DynamoDB assumes that the attribute value does** not exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for Exists is *True*.

The Value and Exists parameters are incompatible with AttributeValueList and ComparisonOperator . Note that if you use both sets of parameters at once, DynamoDB will return a ValidationException exception.

Parameters **return_values** (*string*)–

Use **ReturnValues** if you want to get the item attributes as they appeared before they were updated with the PutItem request. For PutItem, the valid values are:

- **NONE** - If **ReturnValues** is not specified, or if its value is **NONE**, then nothing is returned. (This setting is the default for ReturnValues.)
- **ALL_OLD** - If PutItem overwrote an attribute name-value pair, then the content of the old item is returned.

Parameters

- **return_consumed_capacity** (*string*) – A value that if set to **TOTAL**, the response includes ConsumedCapacity data for tables and indexes. If set to **INDEXES**, the response includes ConsumedCapacity for indexes. If set to **NONE** (the default), ConsumedCapacity is not included in the response.
- **return_item_collection_metrics** (*string*) – A value that if set to **SIZE**, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to **NONE** (the default), no statistics are returned.
- **conditional_operator** (*string*) –

There is a newer parameter available. Use ConditionExpression instead. Note that if you use ConditionalOperator and ConditionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A logical operator to apply to the conditions in the Expected map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit ConditionalOperator, then **AND** is the default.

The operation will succeed only if the entire map evaluates to true.

Parameters condition_expression (*string*) – A condition that must be satisfied in order for a conditional PutItem operation to succeed.

An expression can contain any of the following:

- **Boolean functions:** `attribute_exists` | `attribute_not_exists` | `contains` | `begins_with` These function names are case-sensitive.
- **Comparison operators:** `=` | `<>` | `<` | `>` | `<=` | `>=` | `BETWEEN` | `IN`
- **Logical operators:** `AND` | `OR` | `NOT`

For more information on condition expressions, go to ‘Specifying Conditions’ in the Amazon DynamoDB Developer Guide.

Parameters expression_attribute_names (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using ExpressionAttributeNames:

- **To shorten an attribute name that is very long or unwieldy in an expression.**
- **To create a placeholder for repeating occurrences of an attribute name in an expression.**
- **To prevent special characters in an attribute name from being misinterpreted in an expression.**

Use the **#** character in an expression to dereference an attribute name. For example, consider the following expression:

- `order.customerInfo.LastName = “Smith” OR order.customerInfo.LastName = “Jones”`

Now suppose that you specified the following for ExpressionAttributeNames:

- `{"#name": "order.customerInfo.LastName"}`

The expression can now be simplified as follows:

- `#name = "Smith" OR #name = "Jones"`

For more information on expression attribute names, go to ‘Accessing Item Attributes’ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_values` (*map*) – One or more values that can be substituted in an expression.

Use the `:` (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ “:avail”:{“S”:"Available"}, “:back”:{“S”:"Backordered"}, “:disc”:{“S”:"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, go to ‘Specifying Conditions’ in the Amazon DynamoDB Developer Guide .

query (*table_name*, *key_conditions*, *index_name=None*, *select=None*, *attributes_to_get=None*, *limit=None*, *consistent_read=None*, *query_filter=None*, *conditional_operator=None*, *scan_index_forward=None*, *exclusive_start_key=None*, *return_consumed_capacity=None*, *projection_expression=None*, *filter_expression=None*, *expression_attribute_names=None*, *expression_attribute_values=None*)

A Query operation directly accesses items from a table using the table primary key, or from an index using the index key. You must provide a specific hash key value. You can narrow the scope of the query by using comparison operators on the range key value, or on the index key. You can use the `ScanIndexForward` parameter to get results in forward or reverse order, by range key or by index key.

Queries that do not return results consume the minimum number of read capacity units for that type of read operation.

If the total number of items meeting the query criteria exceeds the result set size limit of 1 MB, the query stops and results are returned to the user with `LastEvaluatedKey` to continue the query in a subsequent operation. Unlike a Scan operation, a Query operation never returns both an empty result set and a `LastEvaluatedKey` . The `LastEvaluatedKey` is only provided if the results exceed 1 MB, or if you have used `Limit` .

You can query a table, a local secondary index, or a global secondary index. For a query on a table or on a local secondary index, you can set `ConsistentRead` to true and obtain a strongly consistent result. Global secondary indexes support eventually consistent reads only, so do not specify `ConsistentRead` when querying a global secondary index.

Parameters

- **table_name** (*string*) – The name of the table containing the requested items.
- **index_name** (*string*) – The name of an index to query. This index can be any local secondary index or global secondary index on the table.
- **select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the case of an index, some or all of the attributes projected into the index.
- **ALL_ATTRIBUTES** - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index DynamoDB will fetch the entire item from the parent table. If the index is configured to project all item

attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.

- ***ALL_PROJECTED_ATTRIBUTES*** - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying *ALL_ATTRIBUTES*.
- ***COUNT*** - Returns the number of matching items, rather than the matching items themselves.
- ***SPECIFIC_ATTRIBUTES*** - Returns only the attributes listed in `AttributesToGet`. This return value is equivalent to specifying `AttributesToGet` without specifying any value for `Select`. If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency. If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither `Select` nor `AttributesToGet` are specified, DynamoDB defaults to *ALL_ATTRIBUTES* when accessing a table, and *ALL_PROJECTED_ATTRIBUTES* when accessing an index. You cannot use both `Select` and `AttributesToGet` together in a single request, unless the value for `Select` is *SPECIFIC_ATTRIBUTES*. (This usage is equivalent to specifying `AttributesToGet` without any value for `Select`.)

Parameters `attributes_to_get` (*list*) –

There is a newer parameter available. Use `ProjectionExpression` instead. Note that if you use `AttributesToGet` and `ProjectionExpression` at the same time, DynamoDB will return a `ValidationException`.

This parameter allows you to retrieve lists or maps; however, it cannot retrieve individual list or map elements.

The names of one or more attributes to retrieve. If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that `AttributesToGet` has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

You cannot use both `AttributesToGet` and `Select` together in a `Query` request, unless the value for `Select` is *SPECIFIC_ATTRIBUTES*. (This usage is equivalent to specifying `AttributesToGet` without any value for `Select`.)

If you query a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB will fetch each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

If you query a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

Parameters

- **`limit`** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in `LastEvaluatedKey` to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in `LastEvaluatedKey` to apply in a subsequent operation to continue the operation. For more information, see [‘Query and Scan’](#) in the Amazon DynamoDB

Developer Guide .

- **consistent_read** (*boolean*) – A value that if set to *True*, then the operation uses strongly consistent reads; otherwise, eventually consistent reads are used.

Strongly consistent reads are not supported on global secondary indexes. If you query a global secondary index with `ConsistentRead` set to *True*, you will receive an error message.

Parameters key_conditions (*map*) – The selection criteria for the query. For a query on a table, you can have conditions only on the table primary key attributes. You must specify the hash key attribute name and value as an *EQ* condition. You can optionally specify a second condition, referring to the range key attribute. If you do not specify a range key condition, all items under the hash key will be fetched and processed. Any filters will be applied after this.

For a query on an index, you can have conditions only on the index key attributes. You must specify the index hash attribute name and value as an *EQ* condition. You can optionally specify a second condition, referring to the index key range attribute.

Each KeyConditions element consists of an attribute name to compare, along with the following:

- **AttributeValueList - One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the `ComparisonOperator` being used. For type `Number`, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see [‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’](http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters). For `Binary`, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions.
- **ComparisonOperator - A comparator for evaluating attributes, for**

example, equals, greater than, less than, and so on. For `KeyConditions`, only the following comparison operators are supported: *EQ* | *LE* | *LT* | *GE* | *GT* | *BEGINS_WITH* | *BETWEEN*. The following are descriptions of these comparison operators.

- **EQ** [Equal. `AttributeValueList` can contain only one `AttributeValue`] of type `String`, `Number`, or `Binary` (not a set type). If an item contains an `AttributeValue` element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`.
- **LE** [Less than or equal. `AttributeValueList` can contain only one] `AttributeValue` element of type `String`, `Number`, or `Binary` (not a set type). If an item contains an `AttributeValue` element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **LT** [Less than. `AttributeValueList` can contain only one] `AttributeValue` of type `String`, `Number`, or `Binary` (not a set type). If an item contains an `AttributeValue` element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **GE** [Greater than or equal. `AttributeValueList` can contain only one] `AttributeValue` element of type `String`, `Number`, or `Binary` (not a set type). If an item contains an `AttributeValue` element of a different type than the one specified in the request, the

value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.

- **GT** [Greater than. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.
- **BEGINS_WITH** [Checks for a prefix. AttributeValueList can contain] only one AttributeValue of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).
- **BETWEEN** [Greater than or equal to the first value, and less than] or equal to the second value. AttributeValueList must contain two AttributeValue elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not compare to `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`.

For usage examples of AttributeValueList and ComparisonOperator, see ‘[Legacy Conditional Parameters](#)’ in the Amazon DynamoDB Developer Guide.

Parameters `query_filter` (*map*) –

There is a newer parameter available. Use FilterExpression instead. Note that if you use QueryFilter and FilterExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A condition that evaluates the query results after the items are read and returns only the desired values.

Query filters are applied after the items are read, so they do not limit the capacity used.

If you specify more than one condition in the QueryFilter map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the ConditionalOperator parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

QueryFilter does not allow key attributes. You cannot define a filter condition on a hash key or range key.

Each QueryFilter element consists of an attribute name to compare, along with the following:

- **AttributeValueList - One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the operator specified in ComparisonOperator. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see ‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions. For information on specifying data types in JSON, see ‘[JSON Data Format](#)’ in the Amazon DynamoDB Developer Guide.
- **ComparisonOperator - A comparator for evaluating attributes.** For example, equals, greater than, less than, etc. The following comparison operators are available: *EQ* | *NE* | *LE* | *LT* | *GE* | *GT* | *NOT_NULL* | *NULL* | *CONTAINS* | *NOT_CONTAINS* | *BEGINS_WITH* | *IN* | *BETWEEN* For complete descriptions of all comparison operators, see ‘[API Condition.html](#)’.

Parameters `conditional_operator` (*string*) –

This parameter does not support lists or maps.

A logical operator to apply to the conditions in the QueryFilter map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit ConditionalOperator, then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Parameters scan_index_forward (*boolean*) – A value that specifies ascending (true) or descending (false) traversal of the index. DynamoDB returns results reflecting the requested order determined by the range key. If the data type is Number, the results are returned in numeric order. For type String, the results are returned in order of ASCII character code values. For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

If ScanIndexForward is not specified, the results are returned in ascending order.

Parameters exclusive_start_key (*map*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for LastEvaluatedKey in the previous operation.

The data type for ExclusiveStartKey must be String, Number or Binary. No set data types are allowed.

Parameters

- **return_consumed_capacity** (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.
- **projection_expression** (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information on projection expressions, go to ‘Accessing Item Attributes’ in the Amazon DynamoDB Developer Guide .

Parameters filter_expression (*string*) – A condition that evaluates the query results after the items are read and returns only the desired values.

The condition you specify is applied to the items queried; any items that do not match the expression are not returned.

Filter expressions are applied after the items are read, so they do not limit the capacity used.

A FilterExpression has the same syntax as a ConditionExpression . For more information on expression syntax, go to **‘Specifying Conditions’** in the Amazon DynamoDB Developer Guide .

Parameters expression_attribute_names (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using ExpressionAttributeNames :

- **To shorten an attribute name that is very long or unwieldy in an** expression.
- **To create a placeholder for repeating occurrences of an attribute** name in an expression.
- **To prevent special characters in an attribute name from being** misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following expression:

- `order.customerInfo.LastName = "Smith" OR order.customerInfo.LastName = "Jones"`

Now suppose that you specified the following for `ExpressionAttributeNames` :

- `{"#name": "order.customerInfo.LastName"}`

The expression can now be simplified as follows:

- `#name = "Smith" OR #name = "Jones"`

For more information on expression attribute names, go to [‘Accessing Item Attributes’](#) in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_values` (*map*) – One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ "avail": {"S": "Available"}, "back": {"S": "Backordered"}, "disc": {"S": "Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, go to [‘Specifying Conditions’](#) in the Amazon DynamoDB Developer Guide .

scan (*table_name*, *attributes_to_get=None*, *limit=None*, *select=None*, *scan_filter=None*, *conditional_operator=None*, *exclusive_start_key=None*, *return_consumed_capacity=None*, *total_segments=None*, *segment=None*, *projection_expression=None*, *filter_expression=None*, *expression_attribute_names=None*, *expression_attribute_values=None*)

The Scan operation returns one or more items and item attributes by accessing every item in the table. To have DynamoDB return fewer items, you can provide a `ScanFilter` operation.

If the total number of scanned items exceeds the maximum data set size limit of 1 MB, the scan stops and results are returned to the user as a `LastEvaluatedKey` value to continue the scan in a subsequent operation. The results also include the number of items exceeding the limit. A scan can result in no table data meeting the filter criteria.

The result set is eventually consistent.

By default, Scan operations proceed sequentially; however, for faster performance on large tables, applications can request a parallel Scan operation by specifying the `Segment` and `TotalSegments` parameters. For more information, see [‘Parallel Scan’](#) in the Amazon DynamoDB Developer Guide .

Parameters

- **table_name** (*string*) – The name of the table containing the requested items.
- **attributes_to_get** (*list*) –

There is a newer parameter available. Use `ProjectionExpression` instead. Note that if you use `AttributesToGet` and `ProjectionExpression` at the same time, DynamoDB will return a `ValidationException` exception.

This parameter allows you to retrieve lists or maps; however, it cannot retrieve individual list or map elements.

The names of one or more attributes to retrieve. If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

Note that AttributesToGet has no effect on provisioned throughput consumption. DynamoDB determines capacity units consumed based on item size, not on the amount of data that is returned to an application.

Parameters

- **limit** (*integer*) – The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in LastEvaluatedKey to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed data set size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in LastEvaluatedKey to apply in a subsequent operation to continue the operation. For more information, see **‘Query and Scan’** in the Amazon DynamoDB Developer Guide .
- **select** (*string*) – The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, or the count of matching items.
- **ALL_ATTRIBUTES** - Returns all of the item attributes.
- **COUNT** - Returns the number of matching items, rather than the matching items themselves.
- **SPECIFIC_ATTRIBUTES** - Returns only the attributes listed in AttributesToGet . This return value is equivalent to specifying AttributesToGet without specifying any value for Select .

If neither Select nor AttributesToGet are specified, DynamoDB defaults to **ALL_ATTRIBUTES**. You cannot use both AttributesToGet and Select together in a single request, unless the value for Select is **SPECIFIC_ATTRIBUTES**. (This usage is equivalent to specifying AttributesToGet without any value for Select .)

Parameters **scan_filter** (*map*) –

There is a newer parameter available. Use FilterExpression instead. Note that if you use ScanFilter and FilterExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A condition that evaluates the scan results and returns only the desired values.

If you specify more than one condition in the ScanFilter map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the ConditionalOperator parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

Each ScanFilter element consists of an attribute name to compare, along with the following:

- **AttributeValueList** - **One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the operator specified in ComparisonOperator . For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see **‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’** . For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions. For information on specifying data types in JSON, see **‘JSON Data Format’** in the Amazon DynamoDB Developer Guide .
- **ComparisonOperator** - **A comparator for evaluating attributes.** For example, equals, greater than, less than, etc. The following comparison operators are available: *EQ* | *NE* | *LE* | *LT* | *GE* | *GT* | *NOT_NULL* | *NULL* | *CONTAINS* | *NOT_CONTAINS* | *BEGINS_WITH* | *IN* | *BETWEEN* For complete descriptions of all comparison operators, see **‘Condition’** .

Parameters **conditional_operator** (*string*) –

There is a newer parameter available. Use ConditionExpression instead. Note that if you use ConditionalOperator and ConditionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A logical operator to apply to the conditions in the ScanFilter map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit ConditionalOperator, then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Parameters exclusive_start_key (*map*) – The primary key of the first item that this operation will evaluate. Use the value that was returned for LastEvaluatedKey in the previous operation.

The data type for ExclusiveStartKey must be String, Number or Binary. No set data types are allowed.

In a parallel scan, a Scan request that includes ExclusiveStartKey must specify the same segment whose previous Scan returned the corresponding value of LastEvaluatedKey.

Parameters

- **return_consumed_capacity** (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.
- **total_segments** (*integer*) – For a parallel Scan request, TotalSegments represents the total number of segments into which the Scan operation will be divided. The value of TotalSegments corresponds to the number of application workers that will perform the parallel scan. For example, if you want to scan a table using four application threads, specify a TotalSegments value of 4.

The value for TotalSegments must be greater than or equal to 1, and less than or equal to 1000000. If you specify a TotalSegments value of 1, the Scan operation will be sequential rather than parallel.

If you specify TotalSegments, you must also specify Segment.

Parameters segment (*integer*) – For a parallel Scan request, Segment identifies an individual segment to be scanned by an application worker.

Segment IDs are zero-based, so the first segment is always 0. For example, if you want to scan a table using four application threads, the first thread specifies a Segment value of 0, the second thread specifies 1, and so on.

The value of LastEvaluatedKey returned from a parallel Scan request must be used as ExclusiveStartKey with the same segment ID in a subsequent Scan operation.

The value for Segment must be greater than or equal to 0, and less than the value provided for TotalSegments.

If you specify Segment, you must also specify TotalSegments.

Parameters projection_expression (*string*) – A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information on projection expressions, go to ‘Accessing Item Attributes’ in the Amazon DynamoDB Developer Guide.

Parameters `filter_expression` (*string*) – A condition that evaluates the scan results and returns only the desired values.

The condition you specify is applied to the items scanned; any items that do not match the expression are not returned.

Parameters `expression_attribute_names` (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using `ExpressionAttributeNames` :

- **To shorten an attribute name that is very long or unwieldy in an** expression.
- **To create a placeholder for repeating occurrences of an attribute** name in an expression.
- **To prevent special characters in an attribute name from being** misinterpreted in an expression.

Use the # character in an expression to dereference an attribute name. For example, consider the following expression:

- `‘order.customerInfo.LastName = “Smith” OR order.customerInfo.LastName = “Jones”‘`

Now suppose that you specified the following for `ExpressionAttributeNames` :

- `{“#name”:“order.customerInfo.LastName”}`

The expression can now be simplified as follows:

- `#name = “Smith” OR #name = “Jones”`

For more information on expression attribute names, go to ‘Accessing Item Attributes’ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_values` (*map*) – One or more values that can be substituted in an expression.

Use the : (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

Available | Backordered | Discontinued

You would first need to specify `ExpressionAttributeValues` as follows:

`‘{“:avail”:{“S”:“Available”}, “:back”:{“S”:“Backordered”}, “:disc”:{“S”:“Discontinued”}}‘`

You could then use these values in an expression, such as this:

ProductStatus IN (:avail, :back, :disc)

For more information on expression attribute values, go to ‘Specifying Conditions’ in the Amazon DynamoDB Developer Guide .

`update_item` (*table_name*, *key*, *attribute_updates=None*, *expected=None*, *conditional_operator=None*, *return_values=None*, *return_consumed_capacity=None*, *return_item_collection_metrics=None*, *update_expression=None*, *condition_expression=None*, *expression_attribute_names=None*, *expression_attribute_values=None*)

Edits an existing item’s attributes, or adds a new item to the table if it does not already exist. You can put, delete, or add attribute values. You can also perform a conditional update (insert a new attribute name-value pair if it doesn’t exist, or replace an existing name-value pair if it has certain expected attribute values).

You can also return the item’s attribute values in the same `UpdateItem` operation using the `ReturnValues` parameter.

Parameters

- **table_name** (*string*) – The name of the table containing the item to update.
- **key** (*map*) – The primary key of the item to be updated. Each element consists of an attribute name and a value for that attribute.

For the primary key, you must provide all of the attributes. For example, with a hash type primary key, you only need to specify the hash attribute. For a hash-and-range type primary key, you must specify both the hash attribute and the range attribute.

Parameters `attribute_updates` (*map*) –

There is a newer parameter available. Use `UpdateExpression` instead. Note that if you use `AttributeUpdates` and `UpdateExpression` at the same time, DynamoDB will return a `ValidationException` exception.

This parameter can be used for modifying top-level attributes; however, it does not support individual list or map elements.

The names of attributes to be modified, the action to perform on each, and the new value for each. If you are updating an attribute that is an index key attribute for any indexes on that table, the attribute type must match the index key type defined in the `AttributesDefinition` of the table description. You can use `UpdateItem` to update any nonkey attributes.

Attribute values cannot be null. String and Binary type attributes must have lengths greater than zero. Set type attributes must not be empty. Requests with empty values will be rejected with a `ValidationException` exception.

Each `AttributeUpdates` element consists of an attribute name to modify, along with the following:

- Value - The new value, if applicable, for this attribute.
- **Action - A value that specifies how to perform the update. This**

action is only valid for an existing attribute whose data type is Number or is a set; do not use *ADD* for other data types. If an item with the specified primary key is found in the table, the following values perform the following actions:

- ***PUT* - Adds the specified attribute to the item. If the attribute** already exists, it is replaced by the new value.
- ***DELETE* - Removes the attribute and its value, if no value is** specified for *DELETE*. The data type of the specified value must match the existing value's data type. If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set $[a,b,c]$ and the *DELETE* action specifies $[a,c]$, then the final attribute value is $[b]$. Specifying an empty set is an error.

- ***ADD* - Adds the specified value to the item, if the attribute does**

not already exist. If the attribute does exist, then the behavior of *ADD* depends on the data type of the attribute:

- * **If the existing attribute is a number, and if Value is also a number,** then Value is mathematically added to the existing attribute. If Value is a negative number, then it is subtracted from the existing attribute. If you use *ADD* to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use *ADD* for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update doesn't have an attribute named `itemcount`, but you decide to *ADD* the number 3 to this attribute anyway. DynamoDB will create the `itemcount` attribute, set its initial value to 0, and finally add 3 to it. The result will be a new `itemcount` attribute, with a value of 3.
- * **If the existing data type is a set, and if Value is also a set, then** Value is appended to the existing set. For example, if the attribute value is the set $[1,2]$, and the *ADD* action specified $[3]$, then the final attribute value is

[1,2,3]. An error occurs if an *ADD* action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, Value must also be a set of strings.

If no item with the specified key is found in the table, the following

values perform the following actions:

- **PUT** - Causes DynamoDB to create a new item with the specified primary key, and then adds the attribute.
- **DELETE** - Nothing happens, because attributes cannot be deleted from a non-existent item. The operation succeeds, but DynamoDB does not create a new item.
- **ADD** - Causes DynamoDB to create an item with the supplied primary key and number (or set of numbers) for the attribute value. The only data types allowed are Number and Number Set.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

Parameters expected (*map*) –

There is a newer parameter available. Use ConditionExpression instead. Note that if you use Expected and ConditionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A map of attribute/condition pairs. Expected provides a conditional block for the UpdateItem operation.

Each element of Expected consists of an attribute name, a comparison operator, and one or more values. DynamoDB compares the attribute with the value(s) you supplied, using the comparison operator. For each Expected element, the result of the evaluation is either true or false.

If you specify more than one element in the Expected map, then by default all of the conditions must evaluate to true. In other words, the conditions are ANDed together. (You can use the ConditionalOperator parameter to OR the conditions instead. If you do this, then at least one of the conditions must evaluate to true, rather than all of them.)

If the Expected map evaluates to true, then the conditional operation succeeds; otherwise, it fails.

Expected contains the following:

- **AttributeValueList - One or more values to evaluate against the** supplied attribute. The number of values in the list depends on the ComparisonOperator being used. For type Number, value comparisons are numeric. String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, *a* is greater than *A*, and *a* is greater than *B*. For a list of code values, see [‘http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters’](http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters). For type Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values, for example when evaluating query expressions.
- **ComparisonOperator - A comparator for evaluating attributes in the**

AttributeValueList . When performing the comparison, DynamoDB uses strongly consistent reads. The following comparison operators are available: *EQ* | *NE* | *LE* | *LT* | *GE* | *GT* | *NOT_NULL* | *NULL* | *CONTAINS* | *NOT_CONTAINS* | *BEGINS_WITH* | *IN* | *BETWEEN* The following are descriptions of each comparison operator.

- **EQ** [Equal. *EQ* is supported for all datatypes, including lists] and maps. AttributeValueList can contain only one AttributeValue element of type String, Number, Binary,

String Set, Number Set, or Binary Set. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >

- **NE** [Not equal. *NE* is supported for all datatypes, including] lists and maps. AttributeValueList can contain only one AttributeValue of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an AttributeValue of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not equal `{"NS": ["6", "2", "1"]}`. >
- **LE** [Less than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **LT** [Less than. AttributeValueList can contain only one] AttributeValue of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **GE** [Greater than or equal. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **GT** [Greater than. AttributeValueList can contain only one] AttributeValue element of type String, Number, or Binary (not a set type). If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, `{"S": "6"}` does not equal `{"N": "6"}`. Also, `{"N": "6"}` does not compare to `{"NS": ["6", "2", "1"]}`. >
- **NOT_NULL** [The attribute exists. *NOT_NULL* is supported for all] datatypes, including lists and maps. This operator tests for the existence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NOT_NULL*, the result is a Boolean true . This result is because the attribute "a" exists; its data type is not relevant to the *NOT_NULL* comparison operator.
- **NULL** [The attribute does not exist. *NULL* is supported for all] datatypes, including lists and maps. This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using *NULL*, the result is a Boolean false . This is because the attribute "a" exists; its data type is not relevant to the *NULL* comparison operator.
- **CONTAINS** [Checks for a subsequence, or value in a set.] AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it finds an exact match with any member of the set. CONTAINS is supported for lists: When evaluating "a CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- **NOT_CONTAINS** [Checks for absence of a subsequence, or absence of a] value in a set. AttributeValueList can contain only one AttributeValue element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it does not find an exact match with any member of the set. NOT_CONTAINS is supported for lists: When evaluating "a NOT CONTAINS b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- **BEGINS_WITH** [Checks for a prefix. AttributeValueList can contain] only one AttributeValue of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type). >
- **IN** [Checks for matching elements within two sets.] AttributeValueList can contain one or more AttributeValue elements of type String, Number, or Binary (not a set type). These attributes are compared against an existing set type attribute of an item. If any elements of the input set are present in the item attribute, the expression evaluates to true.
- **BETWEEN** [Greater than or equal to the first value, and less than] or equal to the second value. AttributeValueList must contain two AttributeValue elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an AttributeValue element of a different type than the one specified in the request, the value does not match. For example, {"S": "6"} does not compare to {"N": "6"}. Also, {"N": "6"} does not compare to {"NS": ["6", "2", "1"]}

For usage examples of AttributeValueList and ComparisonOperator, see '[Legacy Conditional Parameters](#)' in the Amazon DynamoDB Developer Guide.

For backward compatibility with previous DynamoDB releases, the following parameters can be used instead of AttributeValueList and ComparisonOperator:

- Value - A value for DynamoDB to compare with an attribute.
- **Exists** - A Boolean value that causes DynamoDB to evaluate the value

before attempting the conditional operation:

- **If Exists is True, DynamoDB will check to see if that attribute** value already exists in the table. If it is found, then the condition evaluates to true; otherwise the condition evaluate to false.
- **If Exists is False, DynamoDB assumes that the attribute value does** not exist in the table. If in fact the value does not exist, then the assumption is valid and the condition evaluates to true. If the value is found, despite the assumption that it does not exist, the condition evaluates to false.

Note that the default value for Exists is *True*.

The Value and Exists parameters are incompatible with AttributeValueList and ComparisonOperator. Note that if you use both sets of parameters at once, DynamoDB will return a ValidationException exception.

Parameters `conditional_operator` (*string*)–

There is a newer parameter available. Use ConditionExpression instead. Note that if you use ConditionalOperator and ConditionExpression at the same time, DynamoDB will return a ValidationException exception.

This parameter does not support lists or maps.

A logical operator to apply to the conditions in the Expected map:

- **AND** - If all of the conditions evaluate to true, then the entire map evaluates to true.
- **OR** - If at least one of the conditions evaluate to true, then the entire map evaluates to true.

If you omit ConditionalOperator, then AND is the default.

The operation will succeed only if the entire map evaluates to true.

Parameters `return_values` (*string*) –

Use ReturnValues if you want to get the item attributes as they appeared either before or after they were updated. For UpdateItem, the valid values are:

- **NONE** - If ReturnValues is not specified, or if its value is NONE, then nothing is returned. (This setting is the default for ReturnValues.)
- **ALL_OLD** - If UpdateItem overwrote an attribute name-value pair, then the content of the old item is returned.
- **UPDATED_OLD** - The old versions of only the updated attributes are returned.
- **ALL_NEW** - All of the attributes of the new version of the item are returned.
- **UPDATED_NEW** - The new versions of only the updated attributes are returned.

Parameters

- **return_consumed_capacity** (*string*) – A value that if set to *TOTAL*, the response includes ConsumedCapacity data for tables and indexes. If set to *INDEXES*, the response includes ConsumedCapacity for indexes. If set to *NONE* (the default), ConsumedCapacity is not included in the response.
- **return_item_collection_metrics** (*string*) – A value that if set to *SIZE*, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to *NONE* (the default), no statistics are returned.
- **update_expression** (*string*) – An expression that defines one or more attributes to be updated, the action to be performed on them, and new value(s) for them.

The following action values are available for UpdateExpression.

- **SET** - Adds one or more attributes and values to an item. If any of

these attribute already exist, they are replaced by the new values. You can also use SET to add or subtract from an attribute that is of type Number. SET supports the following functions:

- **if_not_exists** (*path, operand*) - if the item does not contain an attribute at the specified path, then *if_not_exists* evaluates to operand; otherwise, it evaluates to path. You can use this function to avoid overwriting an attribute that may already be present in the item.
- **list_append** (*operand, operand*) - evaluates to a list with a new element added to it. You can append the new element to the start or the end of the list by reversing the order of the operands.

These function names are case-sensitive.

- **REMOVE** - Removes one or more attributes from an item.
- **ADD** - Adds the specified value to the item, if the attribute does

not already exist. If the attribute does exist, then the behavior of *ADD* depends on the data type of the attribute:

- **If the existing attribute is a number, and if Value is also a number**, then Value is mathematically added to the existing attribute. If Value is a negative number, then it is subtracted from the existing attribute. If you use *ADD* to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value. Similarly, if you use *ADD* for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses 0 as the initial value. For example, suppose that the item you want to update doesn't have an attribute named `itemcount`, but you decide to *ADD* the number 3 to this attribute anyway. DynamoDB will create the `itemcount` attribute, set its initial value to 0, and finally add 3 to it. The result will be a new `itemcount` attribute in the item, with a value of 3.
- **If the existing data type is a set and if Value is also a set**, then Value is added to the existing set. For example, if the attribute value is the set `[1,2]`, and the *ADD* action specified `[3]`, then the final attribute value is `[1,2,3]`. An error occurs if an *ADD* action is specified for a set attribute and the attribute type specified does not match the existing set type. Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the Value must also be a set of strings.

The *ADD* action only supports Number and set data types. In addition, *ADD* can only be used on top-level attributes, not nested attributes.

- ***DELETE* - Deletes an element from a set. If a set of values is specified**, then those values are subtracted from the old set. For example, if the attribute value was the set `[a,b,c]` and the *DELETE* action specifies `[a,c]`, then the final attribute value is `[b]`. Specifying an empty set is an error. The *DELETE* action only supports Number and set data types. In addition, *DELETE* can only be used on top-level attributes, not nested attributes.

You can have many actions in a single expression, such as the following: `SET a=:value1, b=:value2 DELETE :value3, :value4, :value5`

For more information on update expressions, go to ‘Modifying Items and Attributes’ in the Amazon DynamoDB Developer Guide .

Parameters `condition_expression` (*string*) – A condition that must be satisfied in order for a conditional update to succeed.

An expression can contain any of the following:

- **Boolean functions:** `attribute_exists` | `attribute_not_exists` | `contains` | `begins_with` These function names are case-sensitive.
- **Comparison operators:** `=` | `<` | `<=` | `>` | `>=` | `BETWEEN` | `IN`
- **Logical operators:** `AND` | `OR` | `NOT`

For more information on condition expressions, go to ‘Specifying Conditions’ in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_names` (*map*) – One or more substitution tokens for simplifying complex expressions. The following are some use cases for using `ExpressionAttributeNames` :

- **To shorten an attribute name that is very long or unwieldy in an expression.**
- **To create a placeholder for repeating occurrences of an attribute name in an expression.**
- **To prevent special characters in an attribute name from being misinterpreted in an expression.**

Use the **# character in an expression to dereference an attribute** name. For example, consider the following expression:

- `'order.customerInfo.LastName = "Smith" OR order.customerInfo.LastName = "Jones"'`

Now suppose that you specified the following for `ExpressionAttributeNames` :

- `{ "#name": "order.customerInfo.LastName" }`

The expression can now be simplified as follows:

- `#name = "Smith" OR #name = "Jones"`

For more information on expression attribute names, go to **'Accessing Item Attributes'** in the Amazon DynamoDB Developer Guide .

Parameters `expression_attribute_values` (*map*) – One or more values that can be substituted in an expression.

Use the **: (colon) character in an expression to dereference an** attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

`Available | Backordered | Discontinued`

You would first need to specify `ExpressionAttributeValues` as follows:

`'{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":{"S":"Discontinued"} }'`

You could then use these values in an expression, such as this:

`ProductStatus IN (:avail, :back, :disc)`

For more information on expression attribute values, go to **'Specifying Conditions'** in the Amazon DynamoDB Developer Guide .

update_table (*table_name*, *provisioned_throughput=None*, *global_secondary_index_updates=None*, *attribute_definitions=None*)

Updates the provisioned throughput for the given table, or manages the global secondary indexes on the table.

You can increase or decrease the table's provisioned throughput values within the maximums and minimums listed in the **'Limits'** section in the Amazon DynamoDB Developer Guide .

In addition, you can use `UpdateTable` to add, modify or delete global secondary indexes on the table. For more information, see **'Managing Global Secondary Indexes'** in the Amazon DynamoDB Developer Guide .

The table must be in the *ACTIVE* state for `UpdateTable` to succeed. `UpdateTable` is an asynchronous operation; while executing the operation, the table is in the *UPDATING* state. While the table is in the *UPDATING* state, the table still has the provisioned throughput from before the call. The table's new provisioned throughput settings go into effect when the table returns to the *ACTIVE* state; at that point, the `UpdateTable` operation is complete.

Parameters

- **attribute_definitions** (*list*) – An array of attributes that describe the key schema for the table and indexes. If you are adding a new global secondary index to the table, `AttributeDefinitions` must include the key element(s) of the new index.
- **table_name** (*string*) – The name of the table to be updated.
- **provisioned_throughput** (*dict*) – Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the `UpdateTable` operation.

For current minimum and maximum provisioned throughput values, see **'Limits'** in the Amazon DynamoDB Developer Guide .

Parameters `global_secondary_index_updates` (*list*) –

An array of one or more global secondary indexes for the table. For each index in the array, you can specify one action:

- Create - add a new global secondary index to the table.
- **Update - modify the provisioned throughput settings of an existing** global secondary index.
- Delete - remove a global secondary index from the table.

boto.dynamodb2.exceptions

exception `boto.dynamodb2.exceptions.ConditionalCheckFailedException` (*status*,
reason,
body=None,
**args*)

exception `boto.dynamodb2.exceptions.DynamoDBError`

exception `boto.dynamodb2.exceptions.InternalServerError` (*status*, *reason*,
body=None, **args*)

exception `boto.dynamodb2.exceptions.ItemCollectionSizeLimitExceededException` (*status*,
reason,
body=None,
**args*)

exception `boto.dynamodb2.exceptions.ItemNotFound`

exception `boto.dynamodb2.exceptions.LimitExceededException` (*status*, *reason*,
body=None, **args*)

exception `boto.dynamodb2.exceptions.ProvisionedThroughputExceededException` (*status*,
reason,
body=None,
**args*)

exception `boto.dynamodb2.exceptions.QueryError`

exception `boto.dynamodb2.exceptions.ResourceInUseException` (*status*, *reason*,
body=None, **args*)

exception `boto.dynamodb2.exceptions.ResourceNotFoundException` (*status*, *reason*,
body=None,
**args*)

exception `boto.dynamodb2.exceptions.UnknownFilterTypeError`

exception `boto.dynamodb2.exceptions.UnknownIndexFieldError`

exception `boto.dynamodb2.exceptions.UnknownSchemaFieldError`

exception `boto.dynamodb2.exceptions.ValidationException` (*status*, *reason*,
body=None, **args*)

4.126 EC2

4.126.1 boto.ec2

This module provides an interface to the Elastic Compute Cloud (EC2) service from AWS.

`boto.ec2.connect_to_region` (*region_name*, ***kw_params*)

Given a valid region name, return a `boto.ec2.connection.EC2Connection`. Any additional parameters after the *region_name* are passed on to the connect method of the region object.

Type str

Parameters *region_name* – The name of the region to connect to.

Return type `boto.ec2.connection.EC2Connection` or None

Returns A connection to the given region, or None if an invalid region name is given

`boto.ec2.get_region` (*region_name*, ***kw_params*)

Find and return a `boto.ec2.regioninfo.RegionInfo` object given a region name.

Type str

Param The name of the region.

Return type `boto.ec2.regioninfo.RegionInfo`

Returns The `RegionInfo` object for the given region or None if an invalid region name is provided.

`boto.ec2.regions` (***kw_params*)

Get all available regions for the EC2 service. You may pass any of the arguments accepted by the `EC2Connection` object's constructor as keyword arguments and they will be passed along to the `EC2Connection` object.

Return type list

Returns A list of `boto.ec2.regioninfo.RegionInfo`

4.126.2 boto.ec2.address

class `boto.ec2.address.Address` (*connection=None*, *public_ip=None*, *instance_id=None*)

Represents an EC2 Elastic IP Address

Variables

- **public_ip** – The Elastic IP address.
- **instance_id** – The instance the address is associated with (if any).
- **domain** – Indicates whether the address is a EC2 address or a VPC address (standard/vpc).
- **allocation_id** – The allocation ID for the address (VPC addresses only).
- **association_id** – The association ID for the address (VPC addresses only).
- **network_interface_id** – The network interface (if any) that the address is associated with (VPC addresses only).
- **network_interface_owner_id** – The owner IID (VPC addresses only).
- **private_ip_address** – The private IP address associated with the Elastic IP address (VPC addresses only).

associate (*instance_id=None*, *network_interface_id=None*, *private_ip_address=None*, *allow_reassociation=False*, *dry_run=False*)

Associate this Elastic IP address with a currently running instance. :see: `boto.ec2.connection.EC2Connection.associate_address()`

```
delete (dry_run=False)
    Free up this Elastic IP address. :see: boto.ec2.connection.EC2Connection.
    release_address()

disassociate (dry_run=False)
    Disassociate this Elastic IP address from a currently running instance. :see: boto.ec2.connection.
    EC2Connection.disassociate_address()

endElement (name, value, connection)

release (dry_run=False)
    Free up this Elastic IP address. :see: boto.ec2.connection.EC2Connection.
    release_address()
```

4.126.3 boto.ec2.autoscale

See the *Auto Scaling Reference*.

4.126.4 boto.ec2.blockdevicemapping

```
class boto.ec2.blockdevicemapping.BlockDeviceMapping (connection=None)
    Represents a collection of BlockDeviceTypes when creating ec2 instances.

    Example: dev_sda1 = BlockDeviceType() dev_sda1.size = 100 # change root volume to 100GB in-
    stead of default bdm = BlockDeviceMapping() bdm['/dev/sda1'] = dev_sda1 reservation = image.run(...,
    block_device_map=bdm, ...)

    Parameters connection (boto.ec2.EC2Connection) – Optional connection.
autoscale_build_list_params (params, prefix="")
ec2_build_list_params (params, prefix="")
endElement (name, value, connection)
startElement (name, attrs, connection)

class boto.ec2.blockdevicemapping.BlockDeviceType (connection=None,
                                                    ephemeral_name=None,
                                                    no_device=False,           vol-
                                                    ume_id=None, snapshot_id=None,
                                                    status=None, attach_time=None,
                                                    delete_on_termination=False,
                                                    size=None, volume_type=None,
                                                    iops=None, encrypted=None)

    Represents parameters for a block device.

endElement (name, value, connection)
startElement (name, attrs, connection)

boto.ec2.blockdevicemapping.EBSBlockDeviceType
    alias of boto.ec2.blockdevicemapping.BlockDeviceType
```

4.126.5 boto.ec2.buyreservation

```
class boto.ec2.buyreservation.BuyReservation

    get (params)
```

`get_instance_type` (*params*)

`get_quantity` (*params*)

`get_region` (*params*)

`get_zone` (*params*)

4.126.6 boto.ec2.cloudwatch

See the *CloudWatch Reference*.

4.126.7 boto.ec2.connection

Represents a connection to the EC2 service.

```
class boto.ec2.connection.EC2Connection (aws_access_key_id=None,
                                           aws_secret_access_key=None, is_secure=True,
                                           host=None, port=None, proxy=None,
                                           proxy_port=None, proxy_user=None,
                                           proxy_pass=None, debug=0,
                                           https_connection_factory=None, re-
                                           gion=None, path='/', api_version=None,
                                           security_token=None, validate_certs=True,
                                           profile_name=None)
```

Init method to create a new connection to EC2.

APIVersion = '2014-10-01'

DefaultRegionEndpoint = 'ec2.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.EC2ResponseError*

allocate_address (*domain=None, dry_run=False*)

Allocate a new Elastic IP address and associate it with your account.

Parameters

- **domain** (*string*) – Optional string. If domain is set to “vpc” the address will be allocated to VPC . Will return address object with *allocation_id*.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.address.Address*

Returns The newly allocated Address

assign_private_ip_addresses (*network_interface_id=None, private_ip_addresses=None, secondary_private_ip_address_count=None, allow_reassignment=False, dry_run=False*)

Assigns one or more secondary private IP addresses to a network interface in Amazon VPC.

Parameters

- **network_interface_id** (*string*) – The network interface to which the IP address will be assigned.
- **private_ip_addresses** (*list*) – Assigns the specified IP addresses as secondary IP addresses to the network interface.

- **secondary_private_ip_address_count** (*int*) – The number of secondary IP addresses to assign to the network interface. You cannot specify this parameter when also specifying `private_ip_addresses`.
- **allow_reassignment** (*bool*) – Specifies whether to allow an IP address that is already assigned to another network interface or instance to be reassigned to the specified network interface.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

associate_address (*instance_id=None, public_ip=None, allocation_id=None, network_interface_id=None, private_ip_address=None, allow_reassociation=False, dry_run=False*)

Associate an Elastic IP address with a currently running instance. This requires one of `public_ip` or `allocation_id` depending on if you're associating a VPC address or a plain EC2 address.

When using an Allocation ID, make sure to pass `None` for `public_ip` as EC2 expects a single parameter and if `public_ip` is passed boto will preference that instead of `allocation_id`.

Parameters

- **instance_id** (*string*) – The ID of the instance
- **public_ip** (*string*) – The public IP address for EC2 based allocations.
- **allocation_id** (*string*) – The allocation ID for a VPC-based elastic IP.
- **network_interface_id** (*string*) – The network interface ID to which elastic IP is to be assigned to
- **private_ip_address** (*string*) – The primary or secondary private IP address to associate with the Elastic IP address.
- **allow_reassociation** (*bool*) – Specify this option to allow an Elastic IP address that is already associated with another network interface or instance to be re-associated with the specified instance or interface.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

associate_address_object (*instance_id=None, public_ip=None, allocation_id=None, network_interface_id=None, private_ip_address=None, allow_reassociation=False, dry_run=False*)

Associate an Elastic IP address with a currently running instance. This requires one of `public_ip` or `allocation_id` depending on if you're associating a VPC address or a plain EC2 address.

When using an Allocation ID, make sure to pass `None` for `public_ip` as EC2 expects a single parameter and if `public_ip` is passed boto will preference that instead of `allocation_id`.

Parameters

- **instance_id** (*string*) – The ID of the instance
- **public_ip** (*string*) – The public IP address for EC2 based allocations.
- **allocation_id** (*string*) – The allocation ID for a VPC-based elastic IP.
- **network_interface_id** (*string*) – The network interface ID to which elastic IP is to be assigned to

- **private_ip_address** (*string*) – The primary or secondary private IP address to associate with the Elastic IP address.
- **allow_reassociation** (*bool*) – Specify this option to allow an Elastic IP address that is already associated with another network interface or instance to be re-associated with the specified instance or interface.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `class:boto.ec2.address.Address`

Returns The associated address instance

attach_network_interface (*network_interface_id, instance_id, device_index, dry_run=False*)

Attaches a network interface to an instance.

Parameters

- **network_interface_id** (*str*) – The ID of the network interface to attach.
- **instance_id** (*str*) – The ID of the instance that will be attached to the network interface.
- **device_index** (*int*) – The index of the device for the network interface attachment on the instance.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

attach_volume (*volume_id, instance_id, device, dry_run=False*)

Attach an EBS volume to an EC2 instance.

Parameters

- **volume_id** (*str*) – The ID of the EBS volume to be attached.
- **instance_id** (*str*) – The ID of the EC2 instance to which it will be attached.
- **device** (*str*) – The device on the instance through which the volume will be exposed (e.g. /dev/sdh)
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

authorize_security_group (*group_name=None, src_security_group_name=None, src_security_group_owner_id=None, ip_protocol=None, from_port=None, to_port=None, cidr_ip=None, group_id=None, src_security_group_group_id=None, dry_run=False*)

Add a new rule to an existing security group. You need to pass in either `src_security_group_name` and `src_security_group_owner_id` OR `ip_protocol`, `from_port`, `to_port`, and `cidr_ip`. In other words, either you are authorizing another group or you are authorizing some ip-based rule.

Parameters

- **group_name** (*string*) – The name of the security group you are adding the rule to.
- **src_security_group_name** (*string*) – The name of the security group you are granting access to.
- **src_security_group_owner_id** (*string*) – The ID of the owner of the security group you are granting access to.
- **ip_protocol** (*string*) – Either `tcp` | `udp` | `icmp`
- **from_port** (*int*) – The beginning port number you are enabling

- **to_port** (*int*) – The ending port number you are enabling
- **cidr_ip** (*string or list of strings*) – The CIDR block you are providing access to. See <http://goo.gl/Yj5QC>
- **group_id** (*string*) – ID of the EC2 or VPC security group to modify. This is required for VPC security groups and can be used instead of `group_name` for EC2 security groups.
- **src_security_group_group_id** (*string*) – The ID of the security group you are granting access to. Can be used instead of `src_security_group_name`
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

authorize_security_group_deprecated (*group_name*, *src_security_group_name=None*,
src_security_group_owner_id=None,
ip_protocol=None, *from_port=None*,
to_port=None, *cidr_ip=None*, *dry_run=False*)

NOTE: This method uses the old-style request parameters that did not allow a port to be specified when authorizing a group.

Parameters

- **group_name** (*string*) – The name of the security group you are adding the rule to.
- **src_security_group_name** (*string*) – The name of the security group you are granting access to.
- **src_security_group_owner_id** (*string*) – The ID of the owner of the security group you are granting access to.
- **ip_protocol** (*string*) – Either `tcp` | `udp` | `icmp`
- **from_port** (*int*) – The beginning port number you are enabling
- **to_port** (*string*) – The ending port number you are enabling
- **to_port** – The CIDR block you are providing access to. See <http://goo.gl/Yj5QC>
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

authorize_security_group_egress (*group_id*, *ip_protocol*, *from_port=None*, *to_port=None*,
src_group_id=None, *cidr_ip=None*, *dry_run=False*)

The action adds one or more egress rules to a VPC security group. Specifically, this action permits instances in a security group to send traffic to one or more destination CIDR IP address ranges, or to one or more destination security groups in the same VPC.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

build_configurations_param_list (*params*, *target_configurations*)

build_filter_params (*params*, *filters*)

build_tag_param_list (*params*, *tags*)

bundle_instance (*instance_id, s3_bucket, s3_prefix, s3_upload_policy, dry_run=False*)

Bundle Windows instance.

Parameters

- **instance_id** (*string*) – The instance id
- **s3_bucket** (*string*) – The bucket in which the AMI should be stored.
- **s3_prefix** (*string*) – The beginning of the file name for the AMI.
- **s3_upload_policy** (*string*) – Base64 encoded policy that specifies condition and permissions for Amazon EC2 to upload the user’s image into Amazon S3.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

cancel_bundle_task (*bundle_id, dry_run=False*)

Cancel a previously submitted bundle task

Parameters

- **bundle_id** (*string*) – The identifier of the bundle task to cancel.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

cancel_reserved_instances_listing (*reserved_instances_listing_ids=None, dry_run=False*)

Cancels the specified Reserved Instance listing.

Parameters

- **reserved_instances_listing_ids** (*List of strings*) – The ID of the Reserved Instance listing to be cancelled.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.reservedinstance.ReservedInstanceListing*

cancel_spot_instance_requests (*request_ids, dry_run=False*)

Cancel the specified Spot Instance Requests.

Parameters

- **request_ids** (*list*) – A list of strings of the Request IDs to terminate
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of the instances terminated

confirm_product_instance (*product_code, instance_id, dry_run=False*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

copy_image (*source_region, source_image_id, name=None, description=None, client_token=None, dry_run=False, encrypted=None, kms_key_id=None*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.image.CopyImage*

Returns Object containing the image_id of the copied image.

copy_snapshot (*source_region, source_snapshot_id, description=None, dry_run=False*)

Copies a point-in-time snapshot of an Amazon Elastic Block Store (Amazon EBS) volume and stores it in Amazon Simple Storage Service (Amazon S3). You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create new Amazon EBS volumes or Amazon Machine Images (AMIs).

Parameters

- **source_region** (*str*) – The ID of the AWS region that contains the snapshot to be copied (e.g. 'us-east-1', 'us-west-2', etc.).
- **source_snapshot_id** (*str*) – The ID of the Amazon EBS snapshot to copy
- **description** (*str*) – A description of the new Amazon EBS snapshot.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *str***Returns** The snapshot ID

create_image (*instance_id*, *name*, *description=None*, *no_reboot=False*,
block_device_mapping=None, *dry_run=False*)

Will create an AMI from the instance in the running or stopped state.

Parameters

- **instance_id** (*string*) – the ID of the instance to image.
- **name** (*string*) – The name of the new image
- **description** (*string*) – An optional human-readable string describing the contents and purpose of the AMI.
- **no_reboot** (*bool*) – An optional flag indicating that the bundling process should not attempt to shutdown the instance before bundling. If this flag is True, the responsibility of maintaining file system integrity is left to the owner of the instance.
- **block_device_mapping** (*boto.ec2.blockdevicemapping.BlockDeviceMapping*) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *string***Returns** The new image id

create_key_pair (*key_name*, *dry_run=False*)

Create a new key pair for your account. This will create the key pair within the region you are currently connected to.

Parameters

- **key_name** (*string*) – The name of the new keypair
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.keypair.KeyPair*

Returns The newly created *boto.ec2.keypair.KeyPair*. The material attribute of the new KeyPair object will contain the the unencrypted PEM encoded RSA private key.

create_network_interface (*subnet_id*, *private_ip_address=None*, *description=None*,
groups=None, *dry_run=False*)

Creates a network interface in the specified subnet.

Parameters

- **subnet_id** (*str*) – The ID of the subnet to associate with the network interface.

- **private_ip_address** (*str*) – The private IP address of the network interface. If not supplied, one will be chosen for you.
- **description** (*str*) – The description of the network interface.
- **groups** (*list*) – Lists the groups for use by the network interface. This can be either a list of group ID's or a list of *boto.ec2.securitygroup.SecurityGroup* objects.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.networkinterface.NetworkInterface*

Returns The newly created network interface.

create_placement_group (*name*, *strategy='cluster'*, *dry_run=False*)

Create a new placement group for your account. This will create the placement group within the region you are currently connected to.

Parameters

- **name** (*string*) – The name of the new placement group
- **strategy** (*string*) – The placement strategy of the new placement group. Currently, the only acceptable value is “cluster”.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *bool*

Returns True if successful

create_reserved_instances_listing (*reserved_instances_id*, *instance_count*, *price_schedules*, *client_token*, *dry_run=False*)

Creates a new listing for Reserved Instances.

Creates a new listing for Amazon EC2 Reserved Instances that will be sold in the Reserved Instance Marketplace. You can submit one Reserved Instance listing at a time.

The Reserved Instance Marketplace matches sellers who want to resell Reserved Instance capacity that they no longer need with buyers who want to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances.

If you want to sell your Reserved Instances, you must first register as a Seller in the Reserved Instance Marketplace. After completing the registration process, you can create a Reserved Instance Marketplace listing of some or all of your Reserved Instances, and specify the upfront price you want to receive for them. Your Reserved Instance listings then become available for purchase.

Parameters

- **reserved_instances_id** (*string*) – The ID of the Reserved Instance that will be listed.
- **instance_count** (*int*) – The number of instances that are a part of a Reserved Instance account that will be listed in the Reserved Instance Marketplace. This number should be less than or equal to the instance count associated with the Reserved Instance ID specified in this call.
- **price_schedules** (*List of tuples*) – A list specifying the price of the Reserved Instance for each month remaining in the Reserved Instance term. Each tuple contains two elements, the price and the term. For example, for an instance that 11 months remaining in its term, we can have a price schedule with an upfront price of \$2.50. At 8 months remaining we can drop the price down to \$2.00. This would be expressed as:

```
price_schedules=[('2.50', 11), ('2.00', 8)]
```

- **client_token** (*string*) – Unique, case-sensitive identifier you provide to ensure idempotency of the request. Maximum 64 ASCII characters.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.reservedinstance.ReservedInstanceListing*

create_security_group (*name, description, vpc_id=None, dry_run=False*)

Create a new security group for your account. This will create the security group within the region you are currently connected to.

Parameters

- **name** (*string*) – The name of the new security group
- **description** (*string*) – The description of the new security group
- **vpc_id** (*string*) – The ID of the VPC to create the security group in, if any.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.securitygroup.SecurityGroup*

Returns The newly created *boto.ec2.securitygroup.SecurityGroup*.

create_snapshot (*volume_id, description=None, dry_run=False*)

Create a snapshot of an existing EBS Volume.

Parameters

- **volume_id** (*str*) – The ID of the volume to be snapshot'ed
- **description** (*str*) – A description of the snapshot. Limited to 255 characters.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.snapshot.Snapshot*

Returns The created Snapshot object

create_spot_datafeed_subscription (*bucket, prefix, dry_run=False*)

Create a spot instance datafeed subscription for this account.

Parameters

- **bucket** (*str or unicode*) – The name of the bucket where spot instance data will be written. The account issuing this request must have FULL_CONTROL access to the bucket specified in the request.
- **prefix** (*str or unicode*) – An optional prefix that will be pre-pended to all data files written to the bucket.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.spotdatafeedsubscription.SpotDatafeedSubscription*

Returns The datafeed subscription object or None

create_tags (*resource_ids, tags, dry_run=False*)

Create new metadata tags for the specified resource ids.

Parameters

- **resource_ids** (*list*) – List of strings
- **tags** (*dict*) – A dictionary containing the name/value pairs. If you want to create only a tag name, the value for that tag should be the empty string (e.g. ‘’).
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

create_volume (*size, zone, snapshot=None, volume_type=None, iops=None, encrypted=False, kms_key_id=None, dry_run=False*)

Create a new EBS Volume.

Parameters

- **size** (*int*) – The size of the new volume, in GiB
- **zone** (string or *boto.ec2.zone.Zone*) – The availability zone in which the Volume will be created.
- **snapshot** (string or *boto.ec2.snapshot.Snapshot*) – The snapshot from which the new Volume will be created.
- **volume_type** (*string*) – The type of the volume. (optional). Valid values are: standard | io1 | gp2.
- **iops** (*int*) – The provisioned IOPS you want to associate with this volume. (optional)
- **encrypted** (*bool*) – Specifies whether the volume should be encrypted. (optional)
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Params **kms_key_id** If encrypted is True, this KMS Key ID may be specified to encrypt volume with this key (optional) e.g.: arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef

delete_key_pair (*key_name, dry_run=False*)

Delete a key pair from your account.

Parameters

- **key_name** (*string*) – The name of the keypair to delete
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

delete_network_interface (*network_interface_id, dry_run=False*)

Delete the specified network interface.

Parameters

- **network_interface_id** (*str*) – The ID of the network interface to delete.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

delete_placement_group (*name, dry_run=False*)

Delete a placement group from your account.

Parameters

- **key_name** (*string*) – The name of the keypair to delete
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

delete_security_group (*name=None, group_id=None, dry_run=False*)

Delete a security group from your account.

Parameters

- **name** (*string*) – The name of the security group to delete.
- **group_id** (*string*) – The ID of the security group to delete within a VPC.

- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

delete_snapshot (*snapshot_id, dry_run=False*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

delete_spot_datafeed_subscription (*dry_run=False*)

Delete the current spot instance data feed subscription associated with this account

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_tags (*resource_ids, tags, dry_run=False*)

Delete metadata tags for the specified resource ids.

Parameters

- **resource_ids** (*list*) – List of strings
- **tags** (*dict or list*) – Either a dictionary containing name/value pairs or a list containing just tag names. If you pass in a dictionary, the values must match the actual tag values or the tag will not be deleted. If you pass in a value of None for the tag value, all tags with that name will be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

delete_volume (*volume_id, dry_run=False*)

Delete an EBS volume.

Parameters

- **volume_id** (*str*) – The ID of the volume to be delete.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

deregister_image (*image_id, delete_snapshot=False, dry_run=False*)

Unregister an AMI.

Parameters

- **image_id** (*string*) – the ID of the Image to unregister
- **delete_snapshot** (*bool*) – Set to True if we should delete the snapshot associated with an EBS volume mounted at /dev/sda1
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

describe_account_attributes (*attribute_names=None, dry_run=False*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

describe_reserved_instances_modifications (*reserved_instances_modification_ids=None, next_token=None, filters=None*)

A request to describe the modifications made to Reserved Instances in your account.

Parameters

- **reserved_instances_modification_ids** (*list*) – An optional list of Reserved Instances modification IDs to describe.
- **next_token** (*str*) – A string specifying the next paginated set of results to return.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

Return type `list`

Returns A list of `boto.ec2.reservedinstance.ReservedInstance`

describe_vpc_attribute (*vpc_id, attribute=None, dry_run=False*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

detach_network_interface (*attachment_id, force=False, dry_run=False*)

Detaches a network interface from an instance.

Parameters

- **attachment_id** (*str*) – The ID of the attachment.
- **force** (*bool*) – Set to true to force a detachment.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

detach_volume (*volume_id, instance_id=None, device=None, force=False, dry_run=False*)

Detach an EBS volume from an EC2 instance.

Parameters

- **volume_id** (*str*) – The ID of the EBS volume to be attached.
- **instance_id** (*str*) – The ID of the EC2 instance from which it will be detached.
- **device** (*str*) – The device on the instance through which the volume is exposed (e.g. /dev/sdh)
- **force** (*bool*) – Forces detachment if the previous detachment attempt did not occur cleanly. This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance will not have an opportunity to flush file system caches nor file system meta data. If you use this option, you must perform file system check and repair procedures.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

disassociate_address (*public_ip=None, association_id=None, dry_run=False*)

Disassociate an Elastic IP address from a currently running instance.

Parameters

- **public_ip** (*string*) – The public IP address for EC2 elastic IPs.
- **association_id** (*string*) – The association ID for a VPC based elastic ip.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

enable_volume_io (*volume_id*, *dry_run=False*)

Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.

Parameters

- **volume_id** (*str*) – The ID of the volume.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

get_all_addresses (*addresses=None*, *filters=None*, *allocation_ids=None*, *dry_run=False*)

Get all EIP's associated with the current credentials.

Parameters

- **addresses** (*list*) – Optional list of addresses. If this list is present, only the Addresses associated with these addresses will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **allocation_ids** (*list*) – Optional list of allocation IDs. If this list is present, only the Addresses associated with the given allocation IDs will be returned.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type list of `boto.ec2.address.Address`

Returns The requested Address objects

get_all_bundle_tasks (*bundle_ids=None*, *filters=None*, *dry_run=False*)

Retrieve current bundling tasks. If no bundle id is specified, all tasks are retrieved.

Parameters

- **bundle_ids** (*list*) – A list of strings containing identifiers for previously created bundling tasks.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

get_all_classic_link_instances (*instance_ids=None*, *filters=None*, *dry_run=False*,
max_results=None, *next_token=None*)

Get all of your linked EC2-Classic instances. This request only returns information about EC2-Classic instances linked to a VPC through ClassicLink

Parameters

- **instance_ids** (*list*) – A list of strings of instance IDs. Must be instances linked to a VPC through ClassicLink.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated instance items per response.

Return type *list*

Returns A list of *boto.ec2.instance.Instance*

get_all_images (*image_ids=None, owners=None, executable_by=None, filters=None, dry_run=False*)

Retrieve all the EC2 images available on your account.

Parameters

- **image_ids** (*list*) – A list of strings with the image IDs wanted
- **owners** (*list*) – A list of owner IDs, the special strings ‘self’, ‘amazon’, and ‘aws-marketplace’, may be used to describe images owned by you, Amazon or AWS Marketplace respectively
- **executable_by** (*list*) – Returns AMIs for which the specified user ID has explicit launch permissions
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.image.Image*

get_all_instance_status (*instance_ids=None, max_results=None, next_token=None, filters=None, dry_run=False, include_all_instances=False*)

Retrieve all the instances in your account scheduled for maintenance.

Parameters

- **instance_ids** (*list*) – A list of strings of instance IDs
- **max_results** (*int*) – The maximum number of paginated instance items per response.
- **next_token** (*str*) – A string specifying the next paginated set of results to return.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **include_all_instances** (*bool*) – Set to True if all instances should be returned. (Only running instances are included by default.)

Return type *list*

Returns A list of instances that have maintenance scheduled.

get_all_instance_types ()

Get all instance_types available on this cloud (eucalyptus specific)

Return type *list of boto.ec2.instance_type.InstanceType*

Returns The requested InstanceType objects

get_all_instances (*instance_ids=None, filters=None, dry_run=False, max_results=None*)

Retrieve all the instance reservations associated with your account.

This method's current behavior is deprecated in favor of `get_all_reservations()`. A future major release will change `get_all_instances()` to return a list of `boto.ec2.instance.Instance` objects as its name suggests. To obtain that behavior today, use `get_only_instances()`.

Parameters

- **instance_ids** (*list*) – A list of strings of instance IDs
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated instance items per response.

Return type *list*

Returns A list of `boto.ec2.instance.Reservation`

get_all_kernels (*kernel_ids=None, owners=None, dry_run=False*)

Retrieve all the EC2 kernels available on your account. Constructs a filter to allow the processing to happen server side.

Parameters

- **kernel_ids** (*list*) – A list of strings with the image IDs wanted
- **owners** (*list*) – A list of owner IDs
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of `boto.ec2.image.Image`

get_all_key_pairs (*keynames=None, filters=None, dry_run=False*)

Get all key pairs associated with your account.

Parameters

- **keynames** (*list*) – A list of the names of keypairs to retrieve. If not provided, all key pairs will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of `boto.ec2.keypair.KeyPair`

get_all_network_interfaces (*network_interface_ids=None, filters=None, dry_run=False*)

Retrieve all of the Elastic Network Interfaces (ENI's) associated with your account.

Parameters

- **network_interface_ids** (*list*) – a list of strings representing ENI IDs

- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.networkinterface.NetworkInterface*

get_all_placement_groups (*groupnames=None, filters=None, dry_run=False*)

Get all placement groups associated with your account in a region.

Parameters

- **groupnames** (*list*) – A list of the names of placement groups to retrieve. If not provided, all placement groups will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.placementgroup.PlacementGroup*

get_all_ramdisks (*ramdisk_ids=None, owners=None, dry_run=False*)

Retrieve all the EC2 ramdisks available on your account. Constructs a filter to allow the processing to happen server side.

Parameters

- **ramdisk_ids** (*list*) – A list of strings with the image IDs wanted
- **owners** (*list*) – A list of owner IDs
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.image.Image*

get_all_regions (*region_names=None, filters=None, dry_run=False*)

Get all available regions for the EC2 service.

Parameters

- **region_names** (*list of str*) – Names of regions to limit output
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.regioninfo.RegionInfo*

get_all_reservations (*instance_ids=None, filters=None, dry_run=False, max_results=None, next_token=None*)

Retrieve all the instance reservations associated with your account.

Parameters

- **instance_ids** (*list*) – A list of strings of instance IDs
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated instance items per response.
- **next_token** (*str*) – A string specifying the next paginated set of results to return.

Return type *list*

Returns A list of *boto.ec2.instance.Reservation*

get_all_reserved_instances (*reserved_instances_id=None, filters=None, dry_run=False*)

Describes one or more of the Reserved Instances that you purchased.

Parameters

- **reserved_instance_ids** (*list*) – A list of the reserved instance ids that will be returned. If not provided, all reserved instances will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.reservedinstance.ReservedInstance*

get_all_reserved_instances_offerings (*reserved_instances_offering_ids=None, instance_type=None, availability_zone=None, product_description=None, filters=None, instance_tenancy=None, offering_type=None, include_marketplace=None, min_duration=None, max_duration=None, max_instance_count=None, next_token=None, max_results=None, dry_run=False*)

Describes Reserved Instance offerings that are available for purchase.

Parameters

- **reserved_instances_id** – One or more Reserved Instances offering IDs.
- **instance_type** (*str*) – Displays Reserved Instances of the specified instance type.
- **availability_zone** (*str*) – Displays Reserved Instances within the specified Availability Zone.
- **product_description** (*str*) – Displays Reserved Instances with the specified product description.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

- **instance_tenancy** (*string*) – The tenancy of the Reserved Instance offering. A Reserved Instance with tenancy of dedicated will run on single-tenant hardware and can only be launched within a VPC.
- **offering_type** (*string*) – The Reserved Instance offering type. Valid Values: “*Heavy Utilization*” | “*Medium Utilization*” | “*Light Utilization*”
- **include_marketplace** (*bool*) – Include Marketplace offerings in the response.
- **max_duration** (*int*) – Maximum duration (in seconds) to filter when searching for offerings.
- **max_instance_count** (*int*) – Maximum number of instances to filter when searching for offerings.
- **next_token** (*string*) – Token to use when requesting the next paginated set of offerings.
- **max_results** (*int*) – Maximum number of offerings to return per call.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A *list* of *boto.ec2.reservedinstance.ReservedInstancesOffering*.

get_all_security_groups (*groupnames=None, group_ids=None, filters=None, dry_run=False*)

Get all security groups associated with your account in a region.

Parameters

- **groupnames** (*list*) – A list of the names of security groups to retrieve. If not provided, all security groups will be returned.
- **group_ids** (*list*) – A list of IDs of security groups to retrieve for security groups within a VPC.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.securitygroup.SecurityGroup*

get_all_snapshots (*snapshot_ids=None, owner=None, restorable_by=None, filters=None, dry_run=False*)

Get all EBS Snapshots associated with the current credentials.

Parameters

- **snapshot_ids** (*list*) – Optional list of snapshot ids. If this list is present, only the Snapshots associated with these snapshot ids will be returned.
- **owner** (*str or list*) – If present, only the snapshots owned by the specified user(s) will be returned. Valid values are:
 - self
 - amazon
 - AWS Account ID

- **restorable_by** (*str or list*) – If present, only the snapshots that are restorable by the specified account id(s) will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type list of *boto.ec2.snapshot.Snapshot*

Returns The requested Snapshot objects

get_all_spot_instance_requests (*request_ids=None, filters=None, dry_run=False*)

Retrieve all the spot instances requests associated with your account.

Parameters

- **request_ids** (*list*) – A list of strings of spot instance request IDs
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type list

Returns A list of *boto.ec2.spotinstancerequest.SpotInstanceRequest*

get_all_tags (*filters=None, dry_run=False, max_results=None*)

Retrieve all the metadata tags associated with your account.

Parameters

- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated instance items per response.

Return type list

Returns A list of *boto.ec2.tag.Tag* objects

get_all_volume_status (*volume_ids=None, max_results=None, next_token=None, filters=None, dry_run=False*)

Retrieve the status of one or more volumes.

Parameters

- **volume_ids** (*list*) – A list of strings of volume IDs
- **max_results** (*int*) – The maximum number of paginated instance items per response.
- **next_token** (*str*) – A string specifying the next paginated set of results to return.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the

key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of volume status.

get_all_volumes (*volume_ids=None, filters=None, dry_run=False*)

Get all Volumes associated with the current credentials.

Parameters

- **volume_ids** (*list*) – Optional list of volume ids. If this list is present, only the volumes associated with these volume ids will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list of boto.ec2.volume.Volume*

Returns The requested Volume objects

get_all_zones (*zones=None, filters=None, dry_run=False*)

Get all Availability Zones associated with the current region.

Parameters

- **zones** (*list*) – Optional list of zones. If this list is present, only the Zones associated with these zone names will be returned.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list of boto.ec2.zone.Zone*

Returns The requested Zone objects

get_console_output (*instance_id, dry_run=False*)

Retrieves the console output for the specified instance.

Parameters

- **instance_id** (*string*) – The instance ID of a running instance on the cloud.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.instance.ConsoleOutput*

Returns The console output as a ConsoleOutput object

get_image (*image_id, dry_run=False*)

Shortcut method to retrieve a specific image (AMI).

Parameters

- **image_id** (*string*) – the ID of the Image to retrieve
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.image.Image*

Returns The EC2 Image specified or None if the image is not found

get_image_attribute (*image_id*, *attribute*='launchPermission', *dry_run*=False)

Gets an attribute from an image.

Parameters

- **image_id** (*string*) – The Amazon image id for which you want info about
- **attribute** (*string*) – The attribute you need information about. Valid choices are: * launchPermission * productCodes * blockDeviceMapping
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.image.ImageAttribute*

Returns An ImageAttribute object representing the value of the attribute requested

get_instance_attribute (*instance_id*, *attribute*, *dry_run*=False)

Gets an attribute from an instance.

Parameters

- **instance_id** (*string*) – The Amazon id of the instance
- **attribute** (*string*) – The attribute you need information about Valid choices are:
 - instanceType
 - kernel
 - ramdisk
 - userData
 - disableApiTermination
 - instanceInitiatedShutdownBehavior
 - rootDeviceName
 - blockDeviceMapping
 - productCodes
 - sourceDestCheck
 - groupSet
 - ebsOptimized
 - sriovNetSupport
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.image.InstanceAttribute*

Returns An InstanceAttribute object representing the value of the attribute requested

get_key_pair (*keyname*, *dry_run*=False)

Convenience method to retrieve a specific keypair (KeyPair).

Parameters

- **keyname** (*string*) – The name of the keypair to retrieve
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.keypair.KeyPair*

Returns The KeyPair specified or None if it is not found

get_only_instances (*instance_ids=None, filters=None, dry_run=False, max_results=None*)

Retrieve all the instances associated with your account.

Parameters

- **instance_ids** (*list*) – A list of strings of instance IDs
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated instance items per response.

Return type *list*

Returns A list of *boto.ec2.instance.Instance*

get_params ()

Returns a dictionary containing the value of all of the keyword arguments passed when constructing this connection.

get_password_data (*instance_id, dry_run=False*)

Get encrypted administrator password for a Windows instance.

Parameters

- **instance_id** (*string*) – The identifier of the instance to retrieve the password for.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

get_snapshot_attribute (*snapshot_id, attribute='createVolumePermission', dry_run=False*)

Get information about an attribute of a snapshot. Only one attribute can be specified per call.

Parameters

- **snapshot_id** (*str*) – The ID of the snapshot.
- **attribute** (*str*) – The requested attribute. Valid values are:
 - createVolumePermission
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list of boto.ec2.snapshotattribute.SnapshotAttribute*

Returns The requested Snapshot attribute

get_spot_datafeed_subscription (*dry_run=False*)

Return the current spot instance data feed subscription associated with this account, if any.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.spotdatafeedsubscription.SpotDatafeedSubscription*

Returns The datafeed subscription object or None

get_spot_price_history (*start_time=None, end_time=None, instance_type=None, product_description=None, availability_zone=None, dry_run=False, max_results=None, next_token=None, filters=None*)

Retrieve the recent history of spot instances pricing.

Parameters

- **start_time** (*str*) – An indication of how far back to provide price changes for. An ISO8601 DateTime string.
- **end_time** (*str*) – An indication of how far forward to provide price changes for. An ISO8601 DateTime string.
- **instance_type** (*str*) – Filter responses to a particular instance type.
- **product_description** (*str*) – Filter responses to a particular platform. Valid values are currently:
 - Linux/UNIX
 - SUSE Linux
 - Windows
 - Linux/UNIX (Amazon VPC)
 - SUSE Linux (Amazon VPC)
 - Windows (Amazon VPC)
- **availability_zone** (*str*) – The availability zone for which prices should be returned. If not specified, data for all availability zones will be returned.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **max_results** (*int*) – The maximum number of paginated items per response.
- **next_token** (*str*) – The next set of rows to return. This should be the value of the `next_token` attribute from a previous call to `get_spot_price_history`.
- **filters** (*dict*) – Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

Return type *list*

Returns A list tuples containing price and timestamp.

get_volume_attribute (*volume_id*, *attribute*='autoEnableIO', *dry_run*=False)

Describes attribute of the volume.

Parameters

- **volume_id** (*str*) – The ID of the volume.
- **attribute** (*str*) – The requested attribute. Valid values are:
 - autoEnableIO
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list* of `boto.ec2.volume.VolumeAttribute`

Returns The requested Volume attribute

import_key_pair (*key_name*, *public_key_material*, *dry_run*=False)

imports the public key from an RSA key pair that you created with a third-party tool.

Supported formats:

- OpenSSH public key format (e.g., the format in `~/.ssh/authorized_keys`)
- Base64 encoded DER format
- SSH public key file format as specified in RFC4716

DSA keys are not supported. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 2048, and 4096.

Parameters

- **key_name** (*string*) – The name of the new keypair
- **public_key_material** (*string*) – The public key. You must base64 encode the public key material before sending it to AWS.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.keypair.KeyPair*

Returns A *boto.ec2.keypair.KeyPair* object representing the newly imported key pair. This object will contain only the key name and the fingerprint.

modify_image_attribute (*image_id*, *attribute='launchPermission'*, *operation='add'*, *user_ids=None*, *groups=None*, *product_codes=None*, *dry_run=False*)

Changes an attribute of an image.

Parameters

- **image_id** (*string*) – The image id you wish to change
- **attribute** (*string*) – The attribute you wish to change
- **operation** (*string*) – Either add or remove (this is required for changing launchPermissions)
- **user_ids** (*list*) – The Amazon IDs of users to add/remove attributes
- **groups** (*list*) – The groups to add/remove attributes
- **product_codes** (*list*) – Amazon DevPay product code. Currently only one product code can be associated with an AML. Once set, the product code cannot be changed or reset.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

modify_instance_attribute (*instance_id*, *attribute*, *value*, *dry_run=False*)

Changes an attribute of an instance

Parameters

- **instance_id** (*string*) – The instance id you wish to change
- **attribute** (*string*) – The attribute you wish to change.
 - *instanceType* - A valid instance type (m1.small)
 - *kernel* - Kernel ID (None)
 - *ramdisk* - Ramdisk ID (None)
 - *userData* - Base64 encoded String (None)
 - *disableApiTermination* - Boolean (true)
 - *instanceInitiatedShutdownBehavior* - stop|terminate
 - *blockDeviceMapping* - List of strings - ie: ['/dev/sda=false']
 - *sourceDestCheck* - Boolean (true)
 - *groupSet* - Set of Security Groups or IDs
 - *ebsOptimized* - Boolean (false)

- sriovNetSupport - String - ie: 'simple'
- **value** (*string*) – The new value for the attribute
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns Whether the operation succeeded or not

modify_network_interface_attribute (*interface_id*, *attr*, *value*, *attachment_id=None*, *dry_run=False*)

Changes an attribute of a network interface.

Parameters

- **interface_id** (*string*) – The interface id. Looks like 'eni-xxxxxxx'
- **attr** (*string*) – The attribute you wish to change.
Learn more at <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-ModifyNetworkInterfaceAttribute.html>
 - description - Textual description of interface
 - groupSet - List of security group ids or group objects
 - sourceDestCheck - Boolean
 - deleteOnTermination - Boolean. Must also specify attachment_id
- **value** (*string*) – The new value for the attribute
- **attachment_id** (*string*) – If you're modifying DeleteOnTermination you must specify the attachment_id.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns Whether the operation succeeded or not

modify_reserved_instances (*client_token*, *reserved_instance_ids*, *target_configurations*)

Modifies the specified Reserved Instances.

Parameters

- **client_token** (*string*) – A unique, case-sensitive, token you provide to ensure idempotency of your modification request.
- **reserved_instance_ids** (*List of strings*) – The IDs of the Reserved Instances to modify.
- **target_configurations** (*List of boto.ec2.reservedinstance.ReservedInstancesConfiguration*) – The configuration settings for the modified Reserved Instances.

Return type `string`

Returns The unique ID for the submitted modification request.

modify_snapshot_attribute (*snapshot_id*, *attribute='createVolumePermission'*, *operation='add'*, *user_ids=None*, *groups=None*, *dry_run=False*)

Changes an attribute of an image.

Parameters

- **snapshot_id** (*string*) – The snapshot id you wish to change

- **attribute** (*string*) – The attribute you wish to change. Valid values are: createVolumePermission
- **operation** (*string*) – Either add or remove (this is required for changing snapshot ermissions)
- **user_ids** (*list*) – The Amazon IDs of users to add/remove attributes
- **groups** (*list*) – The groups to add/remove attributes. The only valid value at this time is ‘all’.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

modify_volume_attribute (*volume_id, attribute, new_value, dry_run=False*)

Changes an attribute of an Volume.

Parameters

- **volume_id** (*string*) – The volume id you wish to change
- **attribute** (*string*) – The attribute you wish to change. Valid values are: AutoEnableIO.
- **new_value** (*string*) – The new value of the attribute.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

modify_vpc_attribute (*vpc_id, enable_dns_support=None, enable_dns_hostnames=None, dry_run=False*)

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

monitor_instance (*instance_id, dry_run=False*)

Deprecated Version, maintained for backward compatibility. Enable detailed CloudWatch monitoring for the supplied instance.

Parameters

- **instance_id** (*string*) – The instance id
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.instanceinfo.InstanceInfo*

monitor_instances (*instance_ids, dry_run=False*)

Enable detailed CloudWatch monitoring for the supplied instances.

Parameters

- **instance_id** (*list of strings*) – The instance ids
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.instanceinfo.InstanceInfo*

purchase_reserved_instance_offering (*reserved_instances_offering_id, in-
stance_count=1, limit_price=None,
dry_run=False*)

Purchase a Reserved Instance for use with your account. **** CAUTION **** This request can result in large amounts of money being charged to your AWS account. Use with caution!

Parameters

- **reserved_instances_offering_id** (*string*) – The offering ID of the Reserved Instance to purchase

- **instance_count** (*int*) – The number of Reserved Instances to purchase. Default value is 1.
- **instance_count** – Limit the price on the total order. Must be a tuple of (amount, currency_code), for example: (100.0, 'USD').
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *boto.ec2.reservedinstance.ReservedInstance*

Returns The newly created Reserved Instance

reboot_instances (*instance_ids=None, dry_run=False*)

Reboot the specified instances.

Parameters

- **instance_ids** (*list*) – The instances to terminate and reboot
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

register_image (*name=None, description=None, image_location=None, architecture=None, kernel_id=None, ramdisk_id=None, root_device_name=None, block_device_map=None, dry_run=False, virtualization_type=None, sriov_net_support=None, snapshot_id=None, delete_root_volume_on_termination=False*)

Register an image.

Parameters

- **name** (*string*) – The name of the AMI. Valid only for EBS-based images.
- **description** (*string*) – The description of the AMI.
- **image_location** (*string*) – Full path to your AMI manifest in Amazon S3 storage. Only used for S3-based AMI's.
- **architecture** (*string*) – The architecture of the AMI. Valid choices are: * i386 * x86_64
- **kernel_id** (*string*) – The ID of the kernel with which to launch the instances
- **root_device_name** (*string*) – The root device name (e.g. /dev/sdh)
- **block_device_map** (*boto.ec2.blockdevicemapping.BlockDeviceMapping*) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **virtualization_type** (*string*) – The virtualization_type of the image. Valid choices are: * paravirtual * hvm
- **sriov_net_support** (*string*) – Advanced networking support. Valid choices are: * simple
- **snapshot_id** (*string*) – A snapshot ID for the snapshot to be used as root device for the image. Mutually exclusive with block_device_map, requires root_device_name
- **delete_root_volume_on_termination** (*bool*) – Whether to delete the root volume of the image after instance termination. Only applies when creating image from snapshot_id. Defaults to False. Note that leaving volumes behind after instance termination is not free.

Return type *string*

Returns The new image id

release_address (*public_ip=None, allocation_id=None, dry_run=False*)

Free up an Elastic IP address. Pass a public IP address to release an EC2 Elastic IP address and an AllocationId to release a VPC Elastic IP address. You should only pass one value.

This requires one of `public_ip` or `allocation_id` depending on if you're associating a VPC address or a plain EC2 address.

When using an Allocation ID, make sure to pass `None` for `public_ip` as EC2 expects a single parameter and if `public_ip` is passed boto will preference that instead of `allocation_id`.

Parameters

- **public_ip** (*string*) – The public IP address for EC2 elastic IPs.
- **allocation_id** (*string*) – The Allocation ID for VPC elastic IPs.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

request_spot_instances (*price, image_id, count=1, type='one-time', valid_from=None, valid_until=None, launch_group=None, availability_zone_group=None, key_name=None, security_groups=None, user_data=None, addressing_type=None, instance_type='m1.small', placement=None, kernel_id=None, ramdisk_id=None, monitoring_enabled=False, subnet_id=None, placement_group=None, block_device_map=None, instance_profile_arn=None, instance_profile_name=None, security_group_ids=None, ebs_optimized=False, network_interfaces=None, dry_run=False*)

Request instances on the spot market at a particular price.

Parameters

- **price** (*str*) – The maximum price of your bid
- **image_id** (*string*) – The ID of the image to run
- **count** (*int*) – The of instances to requested
- **type** (*str*) – Type of request. Can be 'one-time' or 'persistent'. Default is one-time.
- **valid_from** (*str*) – Start date of the request. An ISO8601 time string.
- **valid_until** (*str*) – End date of the request. An ISO8601 time string.
- **launch_group** (*str*) – If supplied, all requests will be fulfilled as a group.
- **availability_zone_group** (*str*) – If supplied, all requests will be fulfilled within a single availability zone.
- **key_name** (*string*) – The name of the key pair with which to launch instances
- **security_groups** (*list of strings*) – The names of the security groups with which to associate instances
- **user_data** (*string*) – The user data passed to the launched instances
- **instance_type** (*string*) – The type of instance to run:
 - t1.micro
 - m1.small

- m1.medium
- m1.large
- m1.xlarge
- m3.medium
- m3.large
- m3.xlarge
- m3.2xlarge
- c1.medium
- c1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- cr1.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- cc1.4xlarge
- cg1.4xlarge
- cc2.8xlarge
- g2.2xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- t2.micro
- t2.small
- t2.medium

- **placement** (*string*) – The availability zone in which to launch the instances
- **kernel_id** (*string*) – The ID of the kernel with which to launch the instances
- **ramdisk_id** (*string*) – The ID of the RAM disk with which to launch the instances
- **monitoring_enabled** (*bool*) – Enable detailed CloudWatch monitoring on the instance.
- **subnet_id** (*string*) – The subnet ID within which to launch the instances for VPC.
- **placement_group** (*string*) – If specified, this is the name of the placement group in which the instance(s) will be launched.
- **block_device_map** (*boto.ec2.blockdevicemapping.BlockDeviceMapping*) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.
- **security_group_ids** (*list of strings*) – The ID of the VPC security groups with which to associate instances.
- **instance_profile_arn** (*string*) – The Amazon resource name (ARN) of the IAM Instance Profile (IIP) to associate with the instances.
- **instance_profile_name** (*string*) – The name of the IAM Instance Profile (IIP) to associate with the instances.
- **ebs_optimized** (*bool*) – Whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types.
- **network_interfaces** (*list*) – A list of *boto.ec2.networkinterface.NetworkInterfaceSpecification*
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *Reservation*

Returns The *boto.ec2.spotinstancerequest.SpotInstanceRequest* associated with the request for machines

reset_image_attribute (*image_id, attribute='launchPermission', dry_run=False*)

Resets an attribute of an AMI to its default value.

Parameters

- **image_id** (*string*) – ID of the AMI for which an attribute will be described
- **attribute** (*string*) – The attribute to reset
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *bool*

Returns Whether the operation succeeded or not

reset_instance_attribute (*instance_id, attribute, dry_run=False*)

Resets an attribute of an instance to its default value.

Parameters

- **instance_id** (*string*) – ID of the instance

- **attribute** (*string*) – The attribute to reset. Valid values are: kernellramdisk
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns Whether the operation succeeded or not

reset_snapshot_attribute (*snapshot_id*, *attribute*='createVolumePermission', *dry_run*=False)
Resets an attribute of a snapshot to its default value.

Parameters

- **snapshot_id** (*string*) – ID of the snapshot
- **attribute** (*string*) – The attribute to reset
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns Whether the operation succeeded or not

revoke_security_group (*group_name*=None, *src_security_group_name*=None,
src_security_group_owner_id=None, *ip_protocol*=None,
from_port=None, *to_port*=None, *cidr_ip*=None, *group_id*=None,
src_security_group_group_id=None, *dry_run*=False)

Remove an existing rule from an existing security group. You need to pass in either *src_security_group_name* and *src_security_group_owner_id* OR *ip_protocol*, *from_port*, *to_port*, and *cidr_ip*. In other words, either you are revoking another group or you are revoking some ip-based rule.

Parameters

- **group_name** (*string*) – The name of the security group you are removing the rule from.
- **src_security_group_name** (*string*) – The name of the security group you are revoking access to.
- **src_security_group_owner_id** (*string*) – The ID of the owner of the security group you are revoking access to.
- **ip_protocol** (*string*) – Either tcp | udp | icmp
- **from_port** (*int*) – The beginning port number you are disabling
- **to_port** (*int*) – The ending port number you are disabling
- **cidr_ip** (*string*) – The CIDR block you are revoking access to. See <http://goo.gl/Yj5QC>
- **group_id** (*string*) – ID of the EC2 or VPC security group to modify. This is required for VPC security groups and can be used instead of *group_name* for EC2 security groups.
- **src_security_group_group_id** (*string*) – The ID of the security group for which you are revoking access. Can be used instead of *src_security_group_name*
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

revoke_security_group_deprecated (*group_name*, *src_security_group_name=None*,
src_security_group_owner_id=None,
ip_protocol=None, *from_port=None*, *to_port=None*,
cidr_ip=None, *dry_run=False*)

NOTE: This method uses the old-style request parameters that did not allow a port to be specified when authorizing a group.

Remove an existing rule from an existing security group. You need to pass in either *src_security_group_name* and *src_security_group_owner_id* OR *ip_protocol*, *from_port*, *to_port*, and *cidr_ip*. In other words, either you are revoking another group or you are revoking some ip-based rule.

Parameters

- **group_name** (*string*) – The name of the security group you are removing the rule from.
- **src_security_group_name** (*string*) – The name of the security group you are revoking access to.
- **src_security_group_owner_id** (*string*) – The ID of the owner of the security group you are revoking access to.
- **ip_protocol** (*string*) – Either tcp | udp | icmp
- **from_port** (*int*) – The beginning port number you are disabling
- **to_port** (*string*) – The ending port number you are disabling
- **to_port** – The CIDR block you are revoking access to. <http://goo.gl/Yj5QC>
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

revoke_security_group_egress (*group_id*, *ip_protocol*, *from_port=None*, *to_port=None*,
src_group_id=None, *cidr_ip=None*, *dry_run=False*)

Remove an existing egress rule from an existing VPC security group. You need to pass in an *ip_protocol*, *from_port* and *to_port* range only if the protocol you are using is port-based. You also need to pass in either a *src_group_id* or *cidr_ip*.

Parameters

- **group_id** – The name of the security group you are removing the rule from.
- **ip_protocol** (*string*) – Either tcp | udp | icmp | -1
- **from_port** (*int*) – The beginning port number you are disabling
- **to_port** (*int*) – The ending port number you are disabling
- **src_group_id** (*src_group_id*) – The source security group you are revoking access to.
- **cidr_ip** (*string*) – The CIDR block you are revoking access to. See <http://goo.gl/Yj5QC>
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful.

run_instances (*image_id*, *min_count=1*, *max_count=1*, *key_name=None*, *security_groups=None*, *user_data=None*, *addressing_type=None*, *instance_type='m1.small'*, *placement=None*, *kernel_id=None*, *ramdisk_id=None*, *monitoring_enabled=False*, *subnet_id=None*, *block_device_map=None*, *disable_api_termination=False*, *instance_initiated_shutdown_behavior=None*, *private_ip_address=None*, *placement_group=None*, *client_token=None*, *security_group_ids=None*, *additional_info=None*, *instance_profile_name=None*, *instance_profile_arn=None*, *tenancy=None*, *ebs_optimized=False*, *network_interfaces=None*, *dry_run=False*)

Runs an image on EC2.

Parameters

- **image_id** (*string*) – The ID of the image to run.
- **min_count** (*int*) – The minimum number of instances to launch.
- **max_count** (*int*) – The maximum number of instances to launch.
- **key_name** (*string*) – The name of the key pair with which to launch instances.
- **security_groups** (*list of strings*) – The names of the EC2 classic security groups with which to associate instances
- **user_data** (*string*) – The user data passed to the launched instances
- **instance_type** (*string*) – The type of instance to run:
 - t1.micro
 - m1.small
 - m1.medium
 - m1.large
 - m1.xlarge
 - m3.medium
 - m3.large
 - m3.xlarge
 - m3.2xlarge
 - c1.medium
 - c1.xlarge
 - m2.xlarge
 - m2.2xlarge
 - m2.4xlarge
 - cr1.8xlarge
 - hi1.4xlarge
 - hs1.8xlarge
 - cc1.4xlarge
 - cg1.4xlarge
 - cc2.8xlarge
 - g2.2xlarge

- c3.large
 - c3.xlarge
 - c3.2xlarge
 - c3.4xlarge
 - c3.8xlarge
 - c4.large
 - c4.xlarge
 - c4.2xlarge
 - c4.4xlarge
 - c4.8xlarge
 - i2.xlarge
 - i2.2xlarge
 - i2.4xlarge
 - i2.8xlarge
 - t2.micro
 - t2.small
 - t2.medium
- **placement** (*string*) – The Availability Zone to launch the instance into.
 - **kernel_id** (*string*) – The ID of the kernel with which to launch the instances.
 - **ramdisk_id** (*string*) – The ID of the RAM disk with which to launch the instances.
 - **monitoring_enabled** (*bool*) – Enable detailed CloudWatch monitoring on the instance.
 - **subnet_id** (*string*) – The subnet ID within which to launch the instances for VPC.
 - **private_ip_address** (*string*) – If you’re using VPC, you can optionally use this parameter to assign the instance a specific available IP address from the subnet (e.g., 10.0.0.25).
 - **block_device_map** (*boto.ec2.blockdevicemapping.BlockDeviceMapping*) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.
 - **disable_api_termination** (*bool*) – If True, the instances will be locked and will not be able to be terminated via the API.
 - **instance_initiated_shutdown_behavior** (*string*) – Specifies whether the instance stops or terminates on instance-initiated shutdown. Valid values are:
 - stop
 - terminate

- **placement_group** (*string*) – If specified, this is the name of the placement group in which the instance(s) will be launched.
- **client_token** (*string*) – Unique, case-sensitive identifier you provide to ensure idempotency of the request. Maximum 64 ASCII characters.
- **security_group_ids** (*list of strings*) – The ID of the VPC security groups with which to associate instances.
- **additional_info** (*string*) – Specifies additional information to make available to the instance(s).
- **tenancy** (*string*) – The tenancy of the instance you want to launch. An instance with a tenancy of ‘dedicated’ runs on single-tenant hardware and can only be launched into a VPC. Valid values are:”default” or “dedicated”. NOTE: To use dedicated tenancy you MUST specify a VPC subnet-ID as well.
- **instance_profile_arn** (*string*) – The Amazon resource name (ARN) of the IAM Instance Profile (IIP) to associate with the instances.
- **instance_profile_name** (*string*) – The name of the IAM Instance Profile (IIP) to associate with the instances.
- **ebs_optimized** (*bool*) – Whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn’t available with all instance types.
- **network_interfaces** (*boto.ec2.networkinterface.NetworkInterfaceCollection*) – A NetworkInterfaceCollection data structure containing the ENI specifications for the instance.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *Reservation*

Returns The *boto.ec2.instance.Reservation* associated with the request for machines

start_instances (*instance_ids=None, dry_run=False*)

Start the instances specified

Parameters

- **instance_ids** (*list*) – A list of strings of the Instance IDs to start
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of the instances started

stop_instances (*instance_ids=None, force=False, dry_run=False*)

Stop the instances specified

Parameters

- **instance_ids** (*list*) – A list of strings of the Instance IDs to stop
- **force** (*bool*) – Forces the instance to stop
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of the instances stopped

terminate_instances (*instance_ids=None, dry_run=False*)

Terminate the instances specified

Parameters

- **instance_ids** (*list*) – A list of strings of the Instance IDs to terminate
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of the instances terminated

trim_snapshots (*hourly_backups=8, daily_backups=7, weekly_backups=4, monthly_backups=True*)

Trim excess snapshots, based on when they were taken. More current snapshots are retained, with the number retained decreasing as you move back in time.

If ebs volumes have a ‘Name’ tag with a value, their snapshots will be assigned the same tag when they are created. The values of the ‘Name’ tags for snapshots are used by this function to group snapshots taken from the same volume (or from a series of like-named volumes over time) for trimming.

For every group of like-named snapshots, this function retains the newest and oldest snapshots, as well as, by default, the first snapshots taken in each of the last eight hours, the first snapshots taken in each of the last seven days, the first snapshots taken in the last 4 weeks (counting Midnight Sunday morning as the start of the week), and the first snapshot from the first day of each month forever.

Parameters

- **hourly_backups** (*int*) – How many recent hourly backups should be saved.
- **daily_backups** (*int*) – How many recent daily backups should be saved.
- **weekly_backups** (*int*) – How many recent weekly backups should be saved.
- **monthly_backups** (*int*) – How many monthly backups should be saved. Use True for no limit.

unassign_private_ip_addresses (*network_interface_id=None, private_ip_addresses=None, dry_run=False*)

Unassigns one or more secondary private IP addresses from a network interface in Amazon VPC.

Parameters

- **network_interface_id** (*string*) – The network interface from which the secondary private IP address will be unassigned.
- **private_ip_addresses** (*list*) – Specifies the secondary private IP addresses that you want to unassign from the network interface.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *bool*

Returns True if successful

unmonitor_instance (*instance_id, dry_run=False*)

Deprecated Version, maintained for backward compatibility. Disable detailed CloudWatch monitoring for the supplied instance.

Parameters

- **instance_id** (*string*) – The instance id
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.instanceinfo.InstanceInfo*

unmonitor_instances (*instance_ids*, *dry_run=False*)

Disable CloudWatch monitoring for the supplied instance.

Parameters

- **instance_id** (*list of string*) – The instance id
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.ec2.instanceinfo.InstanceInfo*

4.126.8 boto.ec2.ec2object

Represents an EC2 Object

class `boto.ec2.ec2object.EC2Object` (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.ec2object.TaggedEC2Object` (*connection=None*)

Any EC2 resource that can be tagged should be represented by a Python object that subclasses this class. This class has the mechanism in place to handle the tagSet element in the Describe* responses. If tags are found, it will create a TagSet object and allow it to parse and collect the tags into a dict that is stored in the “tags” attribute of the object.

add_tag (*key, value=”, dry_run=False*)

Add a tag to this object. Tags are stored by AWS and can be used to organize and filter resources. Adding a tag involves a round-trip to the EC2 service.

Parameters

- **key** (*str*) – The key or name of the tag being stored.
- **value** (*str*) – An optional value that can be stored with the tag. If you want only the tag name and no value, the value should be the empty string.

add_tags (*tags, dry_run=False*)

Add tags to this object. Tags are stored by AWS and can be used to organize and filter resources. Adding tags involves a round-trip to the EC2 service.

Parameters tags (*dict*) – A dictionary of key-value pairs for the tags being stored. If for some tags you want only the name and no value, the corresponding value for that tag name should be an empty string.

remove_tag (*key, value=None, dry_run=False*)

Remove a tag from this object. Removing a tag involves a round-trip to the EC2 service.

Parameters

- **key** (*str*) – The key or name of the tag being stored.
- **value** (*str*) – An optional value that can be stored with the tag. If a value is provided, it must match the value currently stored in EC2. If not, the tag will not be removed. If a value of None is provided, the tag will be unconditionally deleted. NOTE: There is an important distinction between a value of “ and a value of None.

remove_tags (*tags, dry_run=False*)

Removes tags from this object. Removing tags involves a round-trip to the EC2 service.

Parameters **tags** (*dict*) – A dictionary of key-value pairs for the tags being removed. For each key, the provided value must match the value currently stored in EC2. If not, that particular tag will not be removed. However, if a value of None is provided, the tag will be unconditionally deleted. NOTE: There is an important distinction between a value of "" and a value of None.

startElement (*name, attrs, connection*)

4.126.9 boto.ec2.elb

See the *ELB Reference*.

4.126.10 boto.ec2.group

class boto.ec2.group.**Group** (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.11 boto.ec2.image

class boto.ec2.image.**BillingProducts**

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.image.**CopyImage** (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.image.**Image** (*connection=None*)

Represents an EC2 Image

deregister (*delete_snapshot=False, dry_run=False*)

endElement (*name, value, connection*)

get_kernel (*dry_run=False*)

get_launch_permissions (*dry_run=False*)

get_ramdisk (*dry_run=False*)

remove_launch_permissions (*user_ids=None, group_names=None, dry_run=False*)

reset_launch_attributes (*dry_run=False*)

run (*min_count=1, max_count=1, key_name=None, security_groups=None, user_data=None, addressing_type=None, instance_type='m1.small', placement=None, kernel_id=None, ramdisk_id=None, monitoring_enabled=False, subnet_id=None, block_device_map=None, disable_api_termination=False, instance_initiated_shutdown_behavior=None, private_ip_address=None, placement_group=None, security_group_ids=None, additional_info=None, instance_profile_name=None, instance_profile_arn=None, tenancy=None, dry_run=False*)

Runs this instance.

Parameters

- **min_count** (*int*) – The minimum number of instances to start
- **max_count** (*int*) – The maximum number of instances to start
- **key_name** (*string*) – The name of the key pair with which to launch instances.
- **security_groups** (*list of strings*) – The names of the security groups with which to associate instances.
- **user_data** (*string*) – The Base64-encoded MIME user data to be made available to the instance(s) in this reservation.
- **instance_type** (*string*) – The type of instance to run:
 - t1.micro
 - m1.small
 - m1.medium
 - m1.large
 - m1.xlarge
 - m3.medium
 - m3.large
 - m3.xlarge
 - m3.2xlarge
 - c1.medium
 - c1.xlarge
 - m2.xlarge
 - m2.2xlarge
 - m2.4xlarge
 - cr1.8xlarge
 - hi1.4xlarge
 - hs1.8xlarge
 - cc1.4xlarge
 - cg1.4xlarge
 - cc2.8xlarge
 - g2.2xlarge
 - c3.large

- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- t2.micro
- t2.small
- t2.medium

- **placement** (*string*) – The Availability Zone to launch the instance into.
- **kernel_id** (*string*) – The ID of the kernel with which to launch the instances.
- **ramdisk_id** (*string*) – The ID of the RAM disk with which to launch the instances.
- **monitoring_enabled** (*bool*) –
Enable CloudWatch monitoring on the instance.

type subnet_id string

- **subnet_id** – The subnet ID within which to launch the instances for VPC.
- **private_ip_address** (*string*) – If you’re using VPC, you can optionally use this parameter to assign the instance a specific available IP address from the subnet (e.g., 10.0.0.25).
- **block_device_map** (*boto.ec2.blockdevicemapping.BlockDeviceMapping*) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.
- **disable_api_termination** (*bool*) – If True, the instances will be locked and will not be able to be terminated via the API.
- **instance_initiated_shutdown_behavior** (*string*) – Specifies whether the instance stops or terminates on instance-initiated shutdown. Valid values are:
 - stop
 - terminate

- **placement_group** (*string*) – If specified, this is the name of the placement group in which the instance(s) will be launched.
- **additional_info** (*string*) – Specifies additional information to make available to the instance(s).
- **security_group_ids** (*list of strings*) – The ID of the VPC security groups with which to associate instances.
- **instance_profile_name** (*string*) – The name of the IAM Instance Profile (IIP) to associate with the instances.
- **instance_profile_arn** (*string*) – The Amazon resource name (ARN) of the IAM Instance Profile (IIP) to associate with the instances.
- **tenancy** (*string*) – The tenancy of the instance you want to launch. An instance with a tenancy of ‘dedicated’ runs on single-tenant hardware and can only be launched into a VPC. Valid values are:”default” or “dedicated”. NOTE: To use dedicated tenancy you MUST specify a VPC subnet-ID as well.

Return type *Reservation*

Returns The *boto.ec2.instance.Reservation* associated with the request for machines

set_launch_permissions (*user_ids=None, group_names=None, dry_run=False*)

startElement (*name, attrs, connection*)

update (*validate=False, dry_run=False*)

Update the image’s state information by making a call to fetch the current image attributes from the service.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the image the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.

class *boto.ec2.image.ImageAttribute* (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class *boto.ec2.image.ProductCodes*

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.12 boto.ec2.instance

Represents an EC2 Instance

class *boto.ec2.instance.ConsoleOutput* (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class *boto.ec2.instance.Instance* (*connection=None*)

Represents an instance.

Variables

- ***id*** – The unique ID of the Instance.
- ***groups*** – A list of Group objects representing the security groups associated with the instance.
- ***public_dns_name*** – The public dns name of the instance.
- ***private_dns_name*** – The private dns name of the instance.
- ***state*** – The string representation of the instance’s current state.
- ***state_code*** – An integer representation of the instance’s current state.
- ***previous_state*** – The string representation of the instance’s previous state.
- ***previous_state_code*** – An integer representation of the instance’s current state.
- ***key_name*** – The name of the SSH key associated with the instance.
- ***instance_type*** – The type of instance (e.g. m1.small).
- ***launch_time*** – The time the instance was launched.
- ***image_id*** – The ID of the AMI used to launch this instance.
- ***placement*** – The availability zone in which the instance is running.
- ***placement_group*** – The name of the placement group the instance is in (for cluster compute instances).
- ***placement_tenancy*** – The tenancy of the instance, if the instance is running within a VPC. An instance with a tenancy of dedicated runs on a single-tenant hardware.
- ***kernel*** – The kernel associated with the instance.
- ***ramdisk*** – The ramdisk associated with the instance.
- ***architecture*** – The architecture of the image (i386|x86_64).
- ***hypervisor*** – The hypervisor used.
- ***virtualization_type*** – The type of virtualization used.
- ***product_codes*** – A list of product codes associated with this instance.
- ***ami_launch_index*** – This instances position within it’s launch group.
- ***monitored*** – A boolean indicating whether monitoring is enabled or not.
- ***monitoring_state*** – A string value that contains the actual value of the monitoring element returned by EC2.
- ***spot_instance_request_id*** – The ID of the spot instance request if this is a spot instance.
- ***subnet_id*** – The VPC Subnet ID, if running in VPC.
- ***vpc_id*** – The VPC ID, if running in VPC.
- ***private_ip_address*** – The private IP address of the instance.
- ***ip_address*** – The public IP address of the instance.
- ***platform*** – Platform of the instance (e.g. Windows)
- ***root_device_name*** – The name of the root device.

- **root_device_type** – The root device type (ebsinstance-store).
- **block_device_mapping** – The Block Device Mapping for the instance.
- **state_reason** – The reason for the most recent state transition.
- **interfaces** – List of Elastic Network Interfaces associated with this instance.
- **ebs_optimized** – Whether instance is using optimized EBS volumes or not.
- **instance_profile** – A Python dict containing the instance profile id and arn associated with this instance.

confirm_product (*product_code*, *dry_run=False*)

create_image (*name*, *description=None*, *no_reboot=False*, *dry_run=False*)

Will create an AMI from the instance in the running or stopped state.

Parameters

- **name** (*string*) – The name of the new image
- **description** (*string*) – An optional human-readable string describing the contents and purpose of the AMI.
- **no_reboot** (*bool*) – An optional flag indicating that the bundling process should not attempt to shutdown the instance before bundling. If this flag is True, the responsibility of maintaining file system integrity is left to the owner of the instance.

Return type string

Returns The new image id

endElement (*name*, *value*, *connection*)

get_attribute (*attribute*, *dry_run=False*)

Gets an attribute from this instance.

Parameters **attribute** (*string*) – The attribute you need information about Valid choices are:

- instanceType
- kernel
- ramdisk
- userData
- disableApiTermination
- instanceInitiatedShutdownBehavior
- rootDeviceName
- blockDeviceMapping
- productCodes
- sourceDestCheck
- groupSet
- ebsOptimized

Return type boto.ec2.image.InstanceAttribute

Returns An InstanceAttribute object representing the value of the attribute requested

get_console_output (*dry_run=False*)

Retrieves the console output for the instance.

Return type *boto.ec2.instance.ConsoleOutput*

Returns The console output as a ConsoleOutput object

modify_attribute (*attribute, value, dry_run=False*)

Changes an attribute of this instance

Parameters

- **attribute** (*string*) – The attribute you wish to change.
 - *instanceType* - A valid instance type (m1.small)
 - *kernel* - Kernel ID (None)
 - *ramdisk* - Ramdisk ID (None)
 - *userData* - Base64 encoded String (None)
 - *disableApiTermination* - Boolean (true)
 - *instanceInitiatedShutdownBehavior* - stop|terminate
 - *sourceDestCheck* - Boolean (true)
 - *groupSet* - Set of Security Groups or IDs
 - *ebsOptimized* - Boolean (false)
- **value** (*string*) – The new value for the attribute

Return type *bool*

Returns Whether the operation succeeded or not

monitor (*dry_run=False*)

placement

placement_group

placement_tenancy

previous_state

previous_state_code

reboot (*dry_run=False*)

reset_attribute (*attribute, dry_run=False*)

Resets an attribute of this instance to its default value.

Parameters **attribute** (*string*) – The attribute to reset. Valid values are: kernelramdisk

Return type *bool*

Returns Whether the operation succeeded or not

start (*dry_run=False*)

Start the instance.

startElement (*name, attrs, connection*)

state

state_code

stop (*force=False, dry_run=False*)

Stop the instance

Parameters **force** (*bool*) – Forces the instance to stop

Return type *list*

Returns A list of the instances stopped

terminate (*dry_run=False*)

Terminate the instance

unmonitor (*dry_run=False*)

update (*validate=False, dry_run=False*)

Update the instance's state information by making a call to fetch the current instance attributes from the service.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the instance the update method returns quietly. If the validate param is True, however, it will raise a `ValueError` exception if no data is returned from EC2.

use_ip (*ip_address, dry_run=False*)

Associates an Elastic IP to the instance.

Parameters **ip_address** (Either an instance of `boto.ec2.address.Address` or a string.) – The IP address to associate with the instance.

Return type *bool*

Returns True if successful

class `boto.ec2.instance.InstanceAttribute` (*parent=None*)

ValidValues = ['instanceType', 'kernel', 'ramdisk', 'userData', 'disableApiTermination']

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.instance.InstancePlacement` (*zone=None, group_name=None, tenancy=None*)

The location where the instance launched.

Variables

- **zone** – The Availability Zone of the instance.
- **group_name** – The name of the placement group the instance is in (for cluster compute instances).
- **tenancy** – The tenancy of the instance (if the instance is running within a VPC). An instance with a tenancy of dedicated runs on single-tenant hardware.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.instance.InstanceState` (*code=0, name=None*)

The state of the instance.

Variables

- **code** – The low byte represents the state. The high byte is an opaque internal value and should be ignored. Valid values:
 - 0 (pending)
 - 16 (running)

- 32 (shutting-down)
- 48 (terminated)
- 64 (stopping)
- 80 (stopped)
- **name** – The name of the state of the instance. Valid values:
 - "pending"
 - "running"
 - "shutting-down"
 - "terminated"
 - "stopping"
 - "stopped"

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.instance.**Reservation** (*connection=None*)

Represents a Reservation response object.

Variables

- **id** – The unique ID of the Reservation.
- **owner_id** – The unique ID of the owner of the Reservation.
- **groups** – A list of Group objects representing the security groups associated with launched instances.
- **instances** – A list of Instance objects launched in this Reservation.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

stop_all (*dry_run=False*)

class boto.ec2.instance.**SubParse** (*section, parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.13 boto.ec2.instanceinfo

class boto.ec2.instanceinfo.**InstanceInfo** (*connection=None, id=None, state=None*)

Represents an EC2 Instance status response from CloudWatch

Variables

- **id** (*str*) – The instance's EC2 ID.
- **state** (*str*) – Specifies the current status of the instance.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.14 boto.ec2.instancestatus

class boto.ec2.instancestatus.Details

A dict object that contains name/value pairs which provide more detailed information about the status of the system or the instance.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.instancestatus.Event (*code=None, description=None, not_before=None, not_after=None*)

A status event for an instance.

Variables

- **code** – A string indicating the event type.
- **description** – A string describing the reason for the event.
- **not_before** – A datestring describing the earliest time for the event.
- **not_after** – A datestring describing the latest time for the event.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.instancestatus.EventSet

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.instancestatus.InstanceStatus (*id=None, zone=None, events=None, state_code=None, state_name=None*)

Represents an EC2 Instance status as reported by DescribeInstanceStatus request.

Variables

- **id** – The instance identifier.
- **zone** – The availability zone of the instance.
- **events** – A list of events relevant to the instance.
- **state_code** – An integer representing the current state of the instance.
- **state_name** – A string describing the current state of the instance.
- **system_status** – A Status object that reports impaired functionality that stems from issues related to the systems that support an instance, such as hardware failures and network connectivity problems.
- **instance_status** – A Status object that reports impaired functionality that arises from problems internal to the instance.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.instancestatus.InstanceStatusSet (*connection=None*)

A list object that contains the results of a call to DescribeInstanceStatus request. Each element of the list will be an InstanceStatus object.

Variables **next_token** – If the response was truncated by the EC2 service, the next_token attribute of the object will contain the string that needs to be passed in to the next request to retrieve the next set of results.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.ec2.instancestatus.Status` (*status=None, details=None*)

A generic Status object used for system status and instance status.

Variables

- **status** – A string indicating overall status.
- **details** – A dict containing name-value pairs which provide more details about the current status.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.15 boto.ec2.keypair

Represents an EC2 Keypair

class `boto.ec2.keypair.KeyPair` (*connection=None*)

copy_to_region (*region, dry_run=False*)

Create a new key pair of the same new in another region. Note that the new key pair will use a different ssh cert than the this key pair. After doing the copy, you will need to save the material associated with the new key pair (use the save method) to a local file.

Parameters **region** (`boto.ec2.regioninfo.RegionInfo`) – The region to which this security group will be copied.

Return type `boto.ec2.keypair.KeyPair`

Returns The new key pair

delete (*dry_run=False*)

Delete the KeyPair.

Return type `bool`

Returns True if successful, otherwise False.

endElement (*name, value, connection*)

save (*directory_path*)

Save the material (the unencrypted PEM encoded RSA private key) of a newly created KeyPair to a local file.

Parameters **directory_path** (*string*) – The fully qualified path to the directory in which the keypair will be saved. The keypair file will be named using the name of the keypair as the base name and .pem for the file extension. If a file of that name already exists in the directory, an exception will be raised and the old file will not be overwritten.

Return type `bool`

Returns True if successful.

4.126.16 boto.ec2.launchspecification

Represents a launch specification for Spot instances.

class `boto.ec2.launchspecification.GroupList`

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.launchspecification.**LaunchSpecification** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.17 boto.ec2.networkinterface

Represents an EC2 Elastic Network Interface

class boto.ec2.networkinterface.**Attachment**

Variables

- **id** – The ID of the attachment.
- **instance_id** – The ID of the instance.
- **device_index** – The index of this device.
- **status** – The status of the device.
- **attach_time** – The time the device was attached.
- **delete_on_termination** – Whether the device will be deleted when the instance is terminated.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.networkinterface.**NetworkInterface** (*connection=None*)

An Elastic Network Interface.

Variables

- **id** – The ID of the ENI.
- **subnet_id** – The ID of the VPC subnet.
- **vpc_id** – The ID of the VPC.
- **description** – The description.
- **owner_id** – The ID of the owner of the ENI.
- **requester_managed** –
- **status** – The interface's status (available/in-use).
- **mac_address** – The MAC address of the interface.
- **private_ip_address** – The IP address of the interface within the subnet.
- **source_dest_check** – Flag to indicate whether to validate network traffic to or from this network interface.
- **groups** – List of security groups associated with the interface.
- **attachment** – The attachment object.
- **private_ip_addresses** – A list of PrivateIPAddress objects.

attach (*instance_id, device_index, dry_run=False*)

Attach this ENI to an EC2 instance.

Parameters

- **instance_id** (*str*) – The ID of the EC2 instance to which it will be attached.
- **device_index** (*int*) – The interface number, N, on the instance (eg. ethN)

Return type `bool`**Returns** True if successful**delete** (*dry_run=False*)**detach** (*force=False, dry_run=False*)

Detach this ENI from an EC2 instance.

Parameters **force** (*bool*) – Forces detachment if the previous detachment attempt did not occur cleanly.**Return type** `bool`**Returns** True if successful**endElement** (*name, value, connection*)**startElement** (*name, attrs, connection*)**update** (*validate=False, dry_run=False*)

Update the data associated with this ENI by querying EC2.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the ENI the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.**class** boto.ec2.networkinterface.**NetworkInterfaceCollection** (**interfaces*)**build_list_params** (*params, prefix=""*)

```
class boto.ec2.networkinterface.NetworkInterfaceSpecification (network_interface_id=None,
device_index=None,
subnet_id=None,
description=None,
private_ip_address=None,
groups=None,
delete_on_termination=None,
private_ip_addresses=None,
secondary_private_ip_address_count=None,
associate_public_ip_address=None)
```

```
class boto.ec2.networkinterface.PrivateIPAddress (connection=None,
private_ip_address=None,
primary=None)
```

endElement (*name, value, connection*)**startElement** (*name, attrs, connection*)

4.126.18 boto.ec2.placementgroup

Represents an EC2 Placement Group

```
class boto.ec2.placementgroup.PlacementGroup (connection=None, name=None, strategy=None, state=None)

    delete (dry_run=False)
    endElement (name, value, connection)
```

4.126.19 boto.ec2.regioninfo

```
class boto.ec2.regioninfo.EC2RegionInfo (connection=None, name=None, endpoint=None, connection_cls=None)

    Represents an EC2 Region
```

4.126.20 boto.ec2.reservedinstance

```
class boto.ec2.reservedinstance.InstanceCount (connection=None, state=None, instance_count=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ModificationResult (connection=None, modification_id=None, availability_zone=None, platform=None, instance_count=None, instance_type=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ModifyReservedInstancesResult (connection=None, modification_id=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.PriceSchedule (connection=None, term=None, price=None, currency_code=None, active=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.PricingDetail (connection=None, price=None, count=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)
```

```

class boto.ec2.reservedinstance.RecurringCharge (connection=None, frequency=None, amount=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ReservedInstance (connection=None, id=None, instance_type=None, availability_zone=None, duration=None, fixed_price=None, usage_price=None, description=None, instance_count=None, state=None)

    endElement (name, value, connection)

class boto.ec2.reservedinstance.ReservedInstanceListing (connection=None, listing_id=None, id=None, create_date=None, update_date=None, status=None, status_message=None, client_token=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ReservedInstancesConfiguration (connection=None, availability_zone=None, platform=None, instance_count=None, instance_type=None)

    endElement (name, value, connection)
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ReservedInstancesModification (connection=None, modification_id=None, reserved_instances=None, modification_results=None, create_date=None, update_date=None, effective_date=None, status=None, status_message=None, client_token=None)

    endElement (name, value, connection)

```

```
    startElement (name, attrs, connection)

class boto.ec2.reservedinstance.ReservedInstancesOffering (connection=None,
                                                            id=None,          in-
                                                            stance_type=None,
                                                            availabil-
                                                            ity_zone=None,
                                                            duration=None,
                                                            fixed_price=None,
                                                            usage_price=None,
                                                            description=None, in-
                                                            stance_tenancy=None,
                                                            currency_code=None,
                                                            offering_type=None,
                                                            recur-
                                                            ring_charges=None,
                                                            pricing_details=None)

    describe ()

    endElement (name, value, connection)

    purchase (instance_count=1, dry_run=False)

    startElement (name, attrs, connection)
```

4.126.21 boto.ec2.securitygroup

Represents an EC2 Security Group

```
class boto.ec2.securitygroup.GroupOrCIDR (parent=None)

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.securitygroup.IPPermissions (parent=None)

    add_grant (name=None, owner_id=None, cidr_ip=None, group_id=None, dry_run=False)

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.securitygroup.IPPermissionsList

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.securitygroup.SecurityGroup (connection=None,          owner_id=None,
                                             name=None, description=None, id=None)

    add_rule (ip_protocol, from_port, to_port, src_group_name, src_group_owner_id, cidr_ip,
              src_group_group_id, dry_run=False)
    Add a rule to the SecurityGroup object. Note that this method only changes the local version of the object.
    No information is sent to EC2.
```

authorize (*ip_protocol=None, from_port=None, to_port=None, cidr_ip=None, src_group=None, dry_run=False*)

Add a new rule to this security group. You need to pass in either `src_group_name` OR `ip_protocol`, `from_port`, `to_port`, and `cidr_ip`. In other words, either you are authorizing another group or you are authorizing some ip-based rule.

Parameters

- **ip_protocol** (*string*) – Either `tcp` | `udp` | `icmp`
- **from_port** (*int*) – The beginning port number you are enabling
- **to_port** (*int*) – The ending port number you are enabling
- **cidr_ip** (*string or list of strings*) – The CIDR block you are providing access to. See http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
- **src_group** (*boto.ec2.securitygroup.SecurityGroup or boto.ec2.securitygroup.GroupOrCIDR*) – The Security Group you are granting access to.

Return type `bool`

Returns True if successful.

copy_to_region (*region, name=None, dry_run=False*)

Create a copy of this security group in another region. Note that the new security group will be a separate entity and will not stay in sync automatically after the copy operation.

Parameters

- **region** (*boto.ec2.regioninfo.RegionInfo*) – The region to which this security group will be copied.
- **name** (*string*) – The name of the copy. If not supplied, the copy will have the same name as this security group.

Return type `boto.ec2.securitygroup.SecurityGroup`

Returns The new security group.

delete (*dry_run=False*)

endElement (*name, value, connection*)

instances (*dry_run=False*)

Find all of the current instances that are running within this security group.

Return type list of `boto.ec2.instance.Instance`

Returns A list of Instance objects

remove_rule (*ip_protocol, from_port, to_port, src_group_name, src_group_owner_id, cidr_ip, src_group_group_id, dry_run=False*)

Remove a rule to the SecurityGroup object. Note that this method only changes the local version of the object. No information is sent to EC2.

revoke (*ip_protocol=None, from_port=None, to_port=None, cidr_ip=None, src_group=None, dry_run=False*)

startElement (*name, attrs, connection*)

4.126.22 boto.ec2.snapshot

Represents an EC2 Elastic Block Store Snapshot

class boto.ec2.snapshot.**Snapshot** (*connection=None*)

Represents an EBS snapshot. :ivar id: The unique ID of the snapshot. :ivar volume_id: The ID of the volume this snapshot was created from. :ivar status: The status of the snapshot. :ivar progress: The percent complete of the snapshot. :ivar start_time: The timestamp of when the snapshot was created. :ivar owner_id: The id of the account that owns the snapshot. :ivar owner_alias: The alias of the account that owns the snapshot. :ivar volume_size: The size (in GB) of the volume the snapshot was created from. :ivar description: The description of the snapshot. :ivar encrypted: True if this snapshot is encrypted

AttrName = 'createVolumePermission'

create_volume (*zone, size=None, volume_type=None, iops=None, dry_run=False*)

Create a new EBS Volume from this Snapshot

Parameters

- **zone** (string or *boto.ec2.zone.Zone*) – The availability zone in which the Volume will be created.
- **size** (*int*) – The size of the new volume, in GiB. (optional). Defaults to the size of the snapshot.
- **volume_type** (*string*) – The type of the volume. (optional). Valid values are: standard | io1 | gp2.
- **iops** (*int*) – The provisioned IOPs you want to associate with this volume. (optional)

delete (*dry_run=False*)

endElement (*name, value, connection*)

get_permissions (*dry_run=False*)

reset_permissions (*dry_run=False*)

share (*user_ids=None, groups=None, dry_run=False*)

unshare (*user_ids=None, groups=None, dry_run=False*)

update (*validate=False, dry_run=False*)

Update the data associated with this snapshot by querying EC2.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the snapshot the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.

class boto.ec2.snapshot.**SnapshotAttribute** (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.23 boto.ec2.spotinstancerequest

Represents an EC2 Spot Instance Request

class boto.ec2.spotinstancerequest.**SpotInstanceRequest** (*connection=None*)

Variables

- **id** – The ID of the Spot Instance Request.
- **price** – The maximum hourly price for any Spot Instance launched to fulfill the request.

- **type** – The Spot Instance request type.
- **state** – The state of the Spot Instance request.
- **fault** – The fault codes for the Spot Instance request, if any.
- **valid_from** – The start date of the request. If this is a one-time request, the request becomes active at this date and time and remains active until all instances launch, the request expires, or the request is canceled. If the request is persistent, the request becomes active at this date and time and remains active until it expires or is canceled.
- **valid_until** – The end date of the request. If this is a one-time request, the request remains active until all instances launch, the request is canceled, or this date is reached. If the request is persistent, it remains active until it is canceled or this date is reached.
- **launch_group** – The instance launch group. Launch groups are Spot Instances that launch together and terminate together.
- **launched_availability_zone** – foo
- **product_description** – The Availability Zone in which the bid is launched.
- **availability_zone_group** – The Availability Zone group. If you specify the same Availability Zone group for all Spot Instance requests, all Spot Instances are launched in the same Availability Zone.
- **create_time** – The time stamp when the Spot Instance request was created.
- **launch_specification** – Additional information for launching instances.
- **instance_id** – The instance ID, if an instance has been launched to fulfill the Spot Instance request.
- **status** – The status code and status message describing the Spot Instance request.

cancel (*dry_run=False*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

```
class boto.ec2.spotinstancerequest.SpotInstanceStateFault (code=None, message=None)
```

The fault codes for the Spot Instance request, if any.

Variables

- **code** – The reason code for the Spot Instance state change.
- **message** – The message for the Spot Instance state change.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

```
class boto.ec2.spotinstancerequest.SpotInstanceStatus (code=None, date_time=None, message=None)
```

Contains the status of a Spot Instance Request.

Variables

- **code** – Status code of the request.
- **message** – The description for the status code for the Spot request.
- **update_time** – Time the status was stated.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.24 boto.ec2.tag

class boto.ec2.tag.**Tag** (*connection=None, res_id=None, res_type=None, name=None, value=None*)

A Tag is used when creating or listing all tags related to an AWS account. It records not only the key and value but also the ID of the resource to which the tag is attached as well as the type of the resource.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.tag.**TagSet** (*connection=None*)

A TagSet is used to collect the tags associated with a particular EC2 resource. Not all resources can be tagged but for those that can, this dict object will be used to collect those values. See [boto.ec2.ec2object.TaggedEC2Object](#) for more details.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.25 boto.ec2.vmtyp

4.126.26 boto.ec2.volume

Represents an EC2 Elastic Block Storage Volume

class boto.ec2.volume.**AttachmentSet**

Represents an EBS attachmentset.

Variables

- **id** – The unique ID of the volume.
- **instance_id** – The unique ID of the attached instance
- **status** – The status of the attachment
- **attach_time** – Attached since
- **device** – The device the instance has mapped

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.volume.**Volume** (*connection=None*)

Represents an EBS volume.

Variables

- **id** – The unique ID of the volume.
- **create_time** – The timestamp of when the volume was created.
- **status** – The status of the volume.
- **size** – The size (in GB) of the volume.
- **snapshot_id** – The ID of the snapshot this volume was created from, if applicable.
- **attach_data** – An AttachmentSet object.
- **zone** – The availability zone this volume is in.
- **type** – The type of volume (standard or consistent-iops)

- **iops** – If this volume is of type consistent-iops, this is the number of IOPS provisioned (10-300).
- **encrypted** – True if this volume is encrypted.

attach (*instance_id, device, dry_run=False*)

Attach this EBS volume to an EC2 instance.

Parameters

- **instance_id** (*str*) – The ID of the EC2 instance to which it will be attached.
- **device** (*str*) – The device on the instance through which the volume will be exposed (e.g. /dev/sdh)

Return type `bool`

Returns True if successful

attachment_state ()

Get the attachment state.

create_snapshot (*description=None, dry_run=False*)

Create a snapshot of this EBS Volume.

Parameters **description** (*str*) – A description of the snapshot. Limited to 256 characters.

Return type `boto.ec2.snapshot.Snapshot`

Returns The created Snapshot object

delete (*dry_run=False*)

Delete this EBS volume.

Return type `bool`

Returns True if successful

detach (*force=False, dry_run=False*)

Detach this EBS volume from an EC2 instance.

Parameters **force** (*bool*) – Forces detachment if the previous detachment attempt did not occur cleanly. This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance will not have an opportunity to flush file system caches nor file system meta data. If you use this option, you must perform file system check and repair procedures.

Return type `bool`

Returns True if successful

endElement (*name, value, connection*)

snapshots (*owner=None, restorable_by=None, dry_run=False*)

Get all snapshots related to this volume. Note that this requires that all available snapshots for the account be retrieved from EC2 first and then the list is filtered client-side to contain only those for this volume.

Parameters

- **owner** (*str*) – If present, only the snapshots owned by the specified user will be returned. Valid values are:
 - self
 - amazon
 - AWS Account ID

- **restorable_by** (*str*) – If present, only the snapshots that are restorable by the specified account id will be returned.

Return type list of L{boto.ec2.snapshot.Snapshot}

Returns The requested Snapshot objects

startElement (*name, attrs, connection*)

update (*validate=False, dry_run=False*)

Update the data associated with this volume by querying EC2.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the volume the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.

volume_state ()

Returns the state of the volume. Same value as the status attribute.

class boto.ec2.volume.**VolumeAttribute** (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.126.27 boto.ec2.volumestatus

class boto.ec2.volumestatus.**Action** (*code=None, id=None, description=None, type=None*)

An action for an instance.

Variables

- **code** – The code for the type of the action.
- **id** – The ID of the event.
- **type** – The type of the event.
- **description** – A description of the action.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.volumestatus.**ActionSet**

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.volumestatus.**Event** (*type=None, id=None, description=None, not_before=None, not_after=None*)

A status event for an instance.

Variables

- **type** – The type of the event.
- **id** – The ID of the event.
- **description** – A string describing the reason for the event.
- **not_before** – A datestring describing the earliest time for the event.
- **not_after** – A datestring describing the latest time for the event.

endElement (*name, value, connection*)

```

    startElement (name, attrs, connection)
class boto.ec2.volumestatus.EventSet

    endElement (name, value, connection)
    startElement (name, attrs, connection)
class boto.ec2.volumestatus.VolumeStatus (id=None, zone=None)
    Represents an EC2 Volume status as reported by DescribeVolumeStatus request.
    Variables
        • id – The volume identifier.
        • zone – The availability zone of the volume
        • volume_status – A Status object that reports impaired functionality that arises
          from problems internal to the instance.
        • events – A list of events relevant to the instance.
        • actions – A list of events relevant to the instance.
    endElement (name, value, connection)
    startElement (name, attrs, connection)
class boto.ec2.volumestatus.VolumeStatusSet (connection=None)
    A list object that contains the results of a call to DescribeVolumeStatus request. Each element of the list will be
    an VolumeStatus object.
    Variables next_token – If the response was truncated by the EC2 service, the next_token at-
    tribute of the object will contain the string that needs to be passed in to the next request to
    retrieve the next set of results.
    endElement (name, value, connection)
    startElement (name, attrs, connection)

```

4.126.28 boto.ec2.zone

Represents an EC2 Availability Zone

```

class boto.ec2.zone.MessageSet
    A list object that contains messages associated with an availability zone.
    endElement (name, value, connection)
    startElement (name, attrs, connection)
class boto.ec2.zone.Zone (connection=None)
    Represents an Availability Zone.
    Variables
        • name – The name of the zone.
        • state – The current state of the zone.
        • region_name – The name of the region the zone is associated with.
        • messages – A list of messages related to the zone.
    endElement (name, value, connection)
    startElement (name, attrs, connection)

```

4.127 EC2 Container Service

4.127.1 boto.ec2containerservice

`boto.ec2containerservice.connect_to_region` (*region_name*, ***kw_params*)

`boto.ec2containerservice.regions` ()

Get all available regions for the Amazon EC2 Container Service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.127.2 boto.ec2containerservice.layer1

class `boto.ec2containerservice.layer1.EC2ContainerServiceConnection` (***kwargs*)

Amazon EC2 Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances. Amazon ECS lets you launch and stop container-enabled applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features like security groups, Amazon EBS volumes, and IAM roles.

You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs, isolation policies, and availability requirements. Amazon EC2 Container Service eliminates the need for you to operate your own cluster management and configuration management systems or worry about scaling your management infrastructure.

APIVersion = '2014-11-13'

DefaultRegionEndpoint = 'ecs.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

create_cluster (*cluster_name=None*)

Creates a new Amazon ECS cluster. By default, your account will receive a *default* cluster when you launch your first container instance. However, you can create your own cluster with a unique name with the *CreateCluster* action.

During the preview, each account is limited to two clusters.

Parameters **cluster_name** (*string*) – The name of your cluster. If you do not specify a name for your cluster, you will create a cluster named *default*.

delete_cluster (*cluster*)

Deletes the specified cluster. You must deregister all container instances from this cluster before you may delete it. You can list the container instances in a cluster with *ListContainerInstances* and deregister them with *DeregisterContainerInstance*.

Parameters **cluster** (*string*) – The cluster you want to delete.

deregister_container_instance (*container_instance*, *cluster=None*, *force=None*)

Deregisters an Amazon ECS container instance from the specified cluster. This instance will no longer be available to run tasks.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instance you want to deregister. If you do not specify a cluster, the default cluster is assumed.

- **container_instance** (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance you want to deregister. The ARN contains the *arn:aws:ecs* namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the *container-instance* namespace, and then the container instance UUID. For example, *arn:aws:ecs: region : aws_account_id :container-instance/ container_instance_UUID*.
- **force** (*boolean*) – Force the deregistration of the container instance. You can use the *force* parameter if you have several tasks running on a container instance and you don't want to run *StopTask* for each task before deregistering the container instance.

deregister_task_definition (*task_definition*)

Deregisters the specified task definition. You will no longer be able to run tasks from this definition after deregistration.

Parameters **task_definition** (*string*) – The *family* and *revision* (*family:revision*) or full Amazon Resource Name (ARN) of the task definition that you want to deregister.

describe_clusters (*clusters=None*)

Describes one or more of your clusters.

Parameters **clusters** (*list*) – A space-separated list of cluster names or full cluster Amazon Resource Name (ARN) entries. If you do not specify a cluster, the default cluster is assumed.

describe_container_instances (*container_instances, cluster=None*)

Describes Amazon EC2 Container Service container instances. Returns metadata about registered and remaining resources on each container instance requested.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances you want to describe. If you do not specify a cluster, the default cluster is assumed.
- **container_instances** (*list*) – A space-separated list of container instance UUIDs or full Amazon Resource Name (ARN) entries.

describe_task_definition (*task_definition*)

Describes a task definition.

Parameters **task_definition** (*string*) – The *family* and *revision* (*family:revision*) or full Amazon Resource Name (ARN) of the task definition that you want to describe.

describe_tasks (*tasks, cluster=None*)

Describes a specified task or tasks.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to describe. If you do not specify a cluster, the default cluster is assumed.
- **tasks** (*list*) – A space-separated list of task UUIDs or full Amazon Resource Name (ARN) entries.

discover_poll_endpoint (*container_instance=None*)

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Returns an endpoint for the Amazon EC2 Container Service agent to poll for updates.

Parameters `container_instance` (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance UUID. For example, `arn:aws:ecs: region : aws_account_id :container- instance/ container_instance_UUID` .

list_clusters (*next_token=None, max_results=None*)

Returns a list of existing clusters.

Parameters

- **next_token** (*string*) – The `nextToken` value returned from a previous paginated `ListClusters` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.
- **max_results** (*integer*) – The maximum number of cluster results returned by `ListClusters` in paginated output. When this parameter is used, `ListClusters` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListClusters` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter is not used, then `ListClusters` returns up to 100 results and a `nextToken` value if applicable.

list_container_instances (*cluster=None, next_token=None, max_results=None*)

Returns a list of container instances in a specified cluster.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances you want to list. If you do not specify a cluster, the default cluster is assumed..
- **next_token** (*string*) – The `nextToken` value returned from a previous paginated `ListContainerInstances` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.
- **max_results** (*integer*) – The maximum number of container instance results returned by `ListContainerInstances` in paginated output. When this parameter is used, `ListContainerInstances` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListContainerInstances` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter is not used, then `ListContainerInstances` returns up to 100 results and a `nextToken` value if applicable.

list_task_definitions (*family_prefix=None, next_token=None, max_results=None*)

Returns a list of task definitions that are registered to your account. You can filter the results by family name with the `familyPrefix` parameter.

Parameters

- **family_prefix** (*string*) – The name of the family that you want to filter the `ListTaskDefinitions` results with. Specifying a `familyPrefix` will limit the listed task definitions to definitions that belong to that family.
- **next_token** (*string*) – The `nextToken` value returned from a previous paginated `ListTaskDefinitions` request where `maxResults` was used and the results ex-

ceeded the value of that parameter. Pagination continues from the end of the previous results that returned the *nextToken* value. This value is *null* when there are no more results to return.

- **max_results** (*integer*) – The maximum number of task definition results returned by *ListTaskDefinitions* in paginated output. When this parameter is used, *ListTaskDefinitions* only returns *maxResults* results in a single page along with a *nextToken* response element. The remaining results of the initial request can be seen by sending another *ListTaskDefinitions* request with the returned *nextToken* value. This value can be between 1 and 100. If this parameter is not used, then *ListTaskDefinitions* returns up to 100 results and a *nextToken* value if applicable.

list_tasks (*cluster=None, container_instance=None, family=None, next_token=None, max_results=None*)

Returns a list of tasks for a specified cluster. You can filter the results by family name or by a particular container instance with the *family* and *containerInstance* parameters.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the tasks you want to list. If you do not specify a cluster, the default cluster is assumed..
- **container_instance** (*string*) – The container instance UUID or full Amazon Resource Name (ARN) of the container instance that you want to filter the *ListTasks* results with. Specifying a *containerInstance* will limit the results to tasks that belong to that container instance.
- **family** (*string*) – The name of the family that you want to filter the *ListTasks* results with. Specifying a *family* will limit the results to tasks that belong to that family.
- **next_token** (*string*) – The *nextToken* value returned from a previous paginated *ListTasks* request where *maxResults* was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the *nextToken* value. This value is *null* when there are no more results to return.
- **max_results** (*integer*) – The maximum number of task results returned by *ListTasks* in paginated output. When this parameter is used, *ListTasks* only returns *maxResults* results in a single page along with a *nextToken* response element. The remaining results of the initial request can be seen by sending another *ListTasks* request with the returned *nextToken* value. This value can be between 1 and 100. If this parameter is not used, then *ListTasks* returns up to 100 results and a *nextToken* value if applicable.

register_container_instance (*cluster=None, instance_identity_document=None, instance_identity_document_signature=None, total_resources=None*)

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Registers an Amazon EC2 instance into the specified cluster. This instance will become available to place containers on.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to register your container instance with. If you do not specify a cluster, the default cluster is assumed..
- **instance_identity_document** (*string*) –

- **instance_identity_document_signature** (*string*) –
- **total_resources** (*list*) –

register_task_definition (*family, container_definitions*)

Registers a new task definition from the supplied *family* and *containerDefinitions*.

Parameters

- **family** (*string*) – You can specify a *family* for a task definition, which allows you to track multiple versions of the same task definition. You can think of the *family* as a name for your task definition.
- **container_definitions** (*list*) – A list of container definitions in JSON format that describe the different containers that make up your task.

run_task (*task_definition, cluster=None, overrides=None, count=None*)

Start a task using random placement and the default Amazon ECS scheduler. If you want to use your own scheduler or place a task on a specific container instance, use *StartTask* instead.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to run your task on. If you do not specify a cluster, the default cluster is assumed..
- **task_definition** (*string*) – The *family* and *revision* (*family:revision*) or full Amazon Resource Name (ARN) of the task definition that you want to run.
- **overrides** (*dict*) –
- **count** (*integer*) – The number of instances of the specified task that you would like to place on your cluster.

start_task (*task_definition, container_instances, cluster=None, overrides=None*)

Starts a new task from the specified task definition on the specified container instance or instances. If you want to use the default Amazon ECS scheduler to place your task, use *RunTask* instead.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that you want to start your task on. If you do not specify a cluster, the default cluster is assumed..
- **task_definition** (*string*) – The *family* and *revision* (*family:revision*) or full Amazon Resource Name (ARN) of the task definition that you want to start.
- **overrides** (*dict*) –
- **container_instances** (*list*) – The container instance UUIDs or full Amazon Resource Name (ARN) entries for the container instances on which you would like to place your task.

stop_task (*task, cluster=None*)

Stops a running task.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task you want to stop. If you do not specify a cluster, the default cluster is assumed..
- **task** (*string*) – The task UUIDs or full Amazon Resource Name (ARN) entry of the task you would like to stop.

submit_container_state_change (*cluster=None, task=None, container_name=None, status=None, exit_code=None, reason=None, network_bindings=None*)

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a container changed states.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container.
- **task** (*string*) – The task UUID or full Amazon Resource Name (ARN) of the task that hosts the container.
- **container_name** (*string*) – The name of the container.
- **status** (*string*) – The status of the state change request.
- **exit_code** (*integer*) – The exit code returned for the state change request.
- **reason** (*string*) – The reason for the state change request.
- **network_bindings** (*list*) – The network bindings of the container.

submit_task_state_change (*cluster=None, task=None, status=None, reason=None*)

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a task changed states.

Parameters

- **cluster** (*string*) – The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task.
- **task** (*string*) – The task UUID or full Amazon Resource Name (ARN) of the task in the state change request.
- **status** (*string*) – The status of the state change request.
- **reason** (*string*) – The reason for the state change request.

4.127.3 boto.ec2containerservice.exceptions

exception boto.ec2containerservice.exceptions.**ClientException** (*status, reason, body=None, *args*)

exception boto.ec2containerservice.exceptions.**ServerException** (*status, reason, body=None, *args*)

4.128 ECS

4.128.1 boto.ecs

```
class boto.ecs.ECSConnection (aws_access_key_id=None,          aws_secret_access_key=None,
                             is_secure=True, port=None, proxy=None, proxy_port=None,
                             proxy_user=None, proxy_pass=None, host='ecs.amazonaws.com',
                             debug=0, https_connection_factory=None, path='/', security_token=None, profile_name=None)
```

ECommerce Connection

For more information on how to use this module see:

<http://blog.coredumped.org/2010/09/search-for-books-on-amazon-using-boto.html>

APIVersion = '2010-11-01'

get_response (action, params, page=0, itemSet=None)
Utility method to handle calls to ECS and parsing of responses.

item_lookup (**params)
Returns items that satisfy the lookup query.

For a full list of parameters, see: <http://s3.amazonaws.com/awsdocs/Associates/2011-08-01/prod-adv-api-dg-2011-08-01.pdf>

item_search (search_index, **params)
Returns items that satisfy the search criteria, including one or more search indices.

For a full list of search terms, :see: <http://docs.amazonwebservices.com/AWSECommerceService/2010-09-01/DG/index.html?ItemSearch.html>

4.128.2 boto.ecs.item

```
class boto.ecs.item.Item (connection=None)
    A single Item
```

Initialize this Item

```
class boto.ecs.item.ItemSet (connection, action, params, page=0)
    A special ResponseGroup that has built-in paging, and only creates new Items on the "Item" tag
```

endElement (name, value, connection)

next ()
Special paging functionality

startElement (name, attrs, connection)

to_xml ()
Override to first fetch everything

```
class boto.ecs.item.ResponseGroup (connection=None, nodename=None)
    A Generic "Response Group", which can be anything from the entire list of Items to specific response elements within an item
```

Initialize this Item

endElement (name, value, connection)

get (name)

```

set (name, value)
startElement (name, attrs, connection)
to_xml ()

```

4.129 Amazon ElastiCache

4.129.1 boto.elasticache

```
boto.elasticache.connect_to_region (region_name, **kw_params)
```

```
boto.elasticache.regions ()
```

Get all available regions for the AWS ElastiCache service.

Return type *list*

Returns A list of `boto.regioninfo.RegionInfo`

4.129.2 boto.elasticache.layer1

```
class boto.elasticache.layer1.ElastiCacheConnection (**kwargs)
```

Amazon ElastiCache Amazon ElastiCache is a web service that makes it easier to set up, operate, and scale a distributed cache in the cloud.

With ElastiCache, customers gain all of the benefits of a high- performance, in-memory cache with far less of the administrative burden of launching and managing a distributed cache. The service makes set-up, scaling, and cluster failure handling much simpler than in a self-managed cache deployment.

In addition, through integration with Amazon CloudWatch, customers get enhanced visibility into the key performance statistics associated with their cache and can receive alarms if a part of their cache runs hot.

```
APIVersion = '2013-06-15'
```

```
DefaultRegionEndpoint = 'elasticache.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
authorize_cache_security_group_ingress (cache_security_group_name,
                                         ec2_security_group_name,
                                         ec2_security_group_owner_id)
```

The `AuthorizeCacheSecurityGroupIngress` operation allows network ingress to a cache security group. Applications using ElastiCache must be running on Amazon EC2, and Amazon EC2 security groups are used as the authorization mechanism. You cannot authorize ingress from an Amazon EC2 security group in one Region to an ElastiCache cluster in another Region.

Parameters

- **cache_security_group_name** (*string*) – The cache security group which will allow network ingress.
- **ec2_security_group_name** (*string*) – The Amazon EC2 security group to be authorized for ingress to the cache security group.
- **ec2_security_group_owner_id** (*string*) – The AWS account number of the Amazon EC2 security group owner. Note that this is not the same thing as an AWS access key ID - you must provide a valid AWS account number for this parameter.

```
create_cache_cluster (cache_cluster_id, num_cache_nodes=None, cache_node_type=None,
                       engine=None, replication_group_id=None, en-
                       gine_version=None, cache_parameter_group_name=None,
                       cache_subnet_group_name=None, cache_security_group_names=None,
                       security_group_ids=None, snapshot_arns=None, pre-
                       ferred_availability_zone=None, preferred_maintenance_window=None,
                       port=None, notification_topic_arn=None,
                       auto_minor_version_upgrade=None)
```

The CreateCacheCluster operation creates a new cache cluster. All nodes in the cache cluster run the same protocol-compliant cache engine software - either Memcached or Redis.

Parameters *cache_cluster_id* (*string*) –

The cache cluster identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 20 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters

- ***replication_group_id*** (*string*) – The replication group to which this cache cluster should belong. If this parameter is specified, the cache cluster will be added to the specified replication group as a read replica; otherwise, the cache cluster will be a standalone primary that is not part of any replication group.
- ***num_cache_nodes*** (*integer*) – The initial number of cache nodes that the cache cluster will have.

For a Memcached cluster, valid values are between 1 and 20. If you need to exceed this limit, please fill out the ElastiCache Limit Increase Request form at [“_”](#).

For Redis, only single-node cache clusters are supported at this time, so the value for this parameter must be 1.

Parameters *cache_node_type* (*string*) – The compute and memory capacity of the nodes in the cache cluster.

Valid values for Memcached:

```
cache.t1.micro | cache.m1.small | cache.m1.medium | cache.m1.large | cache.m1.xlarge |
cache.m3.xlarge | cache.m3.2xlarge | cache.m2.xlarge | cache.m2.2xlarge | cache.m2.4xlarge
| cache.c1.xlarge
```

Valid values for Redis:

```
cache.t1.micro | cache.m1.small | cache.m1.medium | cache.m1.large | cache.m1.xlarge |
cache.m2.xlarge | cache.m2.2xlarge | cache.m2.4xlarge | cache.c1.xlarge
```

For a complete listing of cache node types and specifications, see [“_”](#).

Parameters *engine* (*string*) – The name of the cache engine to be used for this cache cluster.

Valid values for this parameter are:

```
memcached | redis
```

Parameters

- ***engine_version*** (*string*) – The version number of the cache engine to be used for this cluster. To view the supported cache engine versions, use the DescribeCacheEngineVersions operation.
- ***cache_parameter_group_name*** (*string*) – The name of the cache parameter group to associate with this cache cluster. If this argument is omitted, the default cache parameter group for the specified engine will be used.

- **cache_subnet_group_name** (*string*) – The name of the cache subnet group to be used for the cache cluster.

Use this parameter only when you are creating a cluster in an Amazon Virtual Private Cloud (VPC).

Parameters **cache_security_group_names** (*list*) – A list of cache security group names to associate with this cache cluster.

Use this parameter only when you are creating a cluster outside of an Amazon Virtual Private Cloud (VPC).

Parameters **security_group_ids** (*list*) – One or more VPC security groups associated with the cache cluster.

Use this parameter only when you are creating a cluster in an Amazon Virtual Private Cloud (VPC).

Parameters **snapshot_arns** (*list*) – A single-element string list containing an Amazon Resource Name (ARN) that uniquely identifies a Redis RDB snapshot file stored in Amazon S3. The snapshot file will be used to populate the Redis cache in the new cache cluster. The Amazon S3 object name in the ARN cannot contain any commas.

Here is an example of an Amazon S3 ARN: *arn:aws:s3:::my_bucket/snapshot1.rdb*

Note: This parameter is only valid if the *Engine* parameter is *redis*.

Parameters **preferred_availability_zone** (*string*) – The EC2 Availability Zone in which the cache cluster will be created.

All cache nodes belonging to a cache cluster are placed in the preferred availability zone.

Default: System chosen availability zone.

Parameters **preferred_maintenance_window** (*string*) – The weekly time range (in UTC) during which system maintenance can occur.

Example: *sun:05:00-sun:09:00*

Parameters

- **port** (*integer*) – The port number on which each of the cache nodes will accept connections.
- **notification_topic_arn** (*string*) –

The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic to which notifications will be sent.

The Amazon SNS topic owner must be the same as the cache cluster owner.

Parameters **auto_minor_version_upgrade** (*boolean*) – Determines whether minor engine upgrades will be applied automatically to the cache cluster during the maintenance window. A value of *True* allows these upgrades to occur; *False* disables automatic upgrades.

Default: *True*

create_cache_parameter_group (*cache_parameter_group_name*,
cache_parameter_group_family, *description*)

The CreateCacheParameterGroup operation creates a new cache parameter group. A cache parameter group is a collection of parameters that you apply to all of the nodes in a cache cluster.

Parameters

- **cache_parameter_group_name** (*string*) – A user-specified name for the cache parameter group.
- **cache_parameter_group_family** (*string*) – The name of the cache parameter group family the cache parameter group can be used with.

Valid values are: *memcached1.4* | *redis2.6*

Parameters **description** (*string*) – A user-specified description for the cache parameter group.

create_cache_security_group (*cache_security_group_name, description*)

The CreateCacheSecurityGroup operation creates a new cache security group. Use a cache security group to control access to one or more cache clusters.

Cache security groups are only used when you are creating a cluster outside of an Amazon Virtual Private Cloud (VPC). If you are creating a cluster inside of a VPC, use a cache subnet group instead. For more information, see CreateCacheSubnetGroup .

Parameters **cache_security_group_name** (*string*) – A name for the cache security group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters. Must not be the word “Default”.

Example: *mysecuritygroup*

Parameters **description** (*string*) – A description for the cache security group.

create_cache_subnet_group (*cache_subnet_group_name, cache_subnet_group_description, subnet_ids*)

The CreateCacheSubnetGroup operation creates a new cache subnet group.

Use this parameter only when you are creating a cluster in an Amazon Virtual Private Cloud (VPC).

Parameters **cache_subnet_group_name** (*string*) – A name for the cache subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

Example: *mysubnetgroup*

Parameters

- **cache_subnet_group_description** (*string*) – A description for the cache subnet group.
- **subnet_ids** (*list*) – A list of VPC subnet IDs for the cache subnet group.

create_replication_group (*replication_group_id, primary_cluster_id, replication_group_description*)

The CreateReplicationGroup operation creates a replication group. A replication group is a collection of cache clusters, where one of the clusters is a read/write primary and the other clusters are read-only replicas. Writes to the primary are automatically propagated to the replicas.

When you create a replication group, you must specify an existing cache cluster that is in the primary role. When the replication group has been successfully created, you can add one or more read replica replicas to it, up to a total of five read replicas.

Parameters **replication_group_id** (*string*) –

The replication group identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 20 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters

- **primary_cluster_id** (*string*) – The identifier of the cache cluster that will serve as the primary for this replication group. This cache cluster must already exist and have a status of available .
- **replication_group_description** (*string*) – A user-specified description for the replication group.

delete_cache_cluster (*cache_cluster_id*)

The DeleteCacheCluster operation deletes a previously provisioned cache cluster. DeleteCacheCluster deletes all associated cache nodes, node endpoints and the cache cluster itself. When you receive a successful response from this operation, Amazon ElastiCache immediately begins deleting the cache cluster; you cannot cancel or revert this operation.

Parameters **cache_cluster_id** (*string*) – The cache cluster identifier for the cluster to be deleted. This parameter is not case sensitive.

delete_cache_parameter_group (*cache_parameter_group_name*)

The DeleteCacheParameterGroup operation deletes the specified cache parameter group. You cannot delete a cache parameter group if it is associated with any cache clusters.

Parameters **cache_parameter_group_name** (*string*) – The name of the cache parameter group to delete.

The specified cache security group must not be associated with any cache clusters.

delete_cache_security_group (*cache_security_group_name*)

The DeleteCacheSecurityGroup operation deletes a cache security group. You cannot delete a cache security group if it is associated with any cache clusters.

Parameters **cache_security_group_name** (*string*) – The name of the cache security group to delete.

You cannot delete the default security group.

delete_cache_subnet_group (*cache_subnet_group_name*)

The DeleteCacheSubnetGroup operation deletes a cache subnet group. You cannot delete a cache subnet group if it is associated with any cache clusters.

Parameters **cache_subnet_group_name** (*string*) – The name of the cache subnet group to delete.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

delete_replication_group (*replication_group_id*)

The DeleteReplicationGroup operation deletes an existing replication group. DeleteReplicationGroup deletes the primary cache cluster and all of the read replicas in the replication group. When you receive a successful response from this operation, Amazon ElastiCache immediately begins deleting the entire replication group; you cannot cancel or revert this operation.

Parameters **replication_group_id** (*string*) – The identifier for the replication group to be deleted. This parameter is not case sensitive.

describe_cache_clusters (*cache_cluster_id=None, max_records=None, marker=None, show_cache_node_info=None*)

The DescribeCacheClusters operation returns information about all provisioned cache clusters if no cache cluster identifier is specified, or about a specific cache cluster if a cache cluster identifier is supplied.

By default, abbreviated information about the cache clusters(s) will be returned. You can use the optional ShowDetails flag to retrieve detailed information about the cache nodes associated with the cache clusters. These details include the DNS address and port for the cache node endpoint.

If the cluster is in the CREATING state, only cluster level information will be displayed until all of the nodes are successfully provisioned.

If the cluster is in the DELETING state, only cluster level information will be displayed.

If cache nodes are currently being added to the cache cluster, node endpoint information and creation time for the additional nodes will not be displayed until they are completely provisioned. When the cache cluster state is available , the cluster is ready for use.

If cache nodes are currently being removed from the cache cluster, no endpoint information for the removed nodes is displayed.

Parameters

- **cache_cluster_id** (*string*) – The user-supplied cluster identifier. If this parameter is specified, only information about that specific cache cluster is returned. This parameter isn't case sensitive.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters

- **marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .
- **show_cache_node_info** (*boolean*) – An optional flag that can be included in the *DescribeCacheCluster* request to retrieve information about the individual cache nodes.

describe_cache_engine_versions (*engine=None, engine_version=None, cache_parameter_group_family=None, max_records=None, marker=None, default_only=None*)

The *DescribeCacheEngineVersions* operation returns a list of the available cache engines and their versions.

Parameters

- **engine** (*string*) – The cache engine to return. Valid values: *memcached* | *redis*
- **engine_version** (*string*) – The cache engine version to return.

Example: *1.4.14*

Parameters **cache_parameter_group_family** (*string*) –

The name of a specific cache parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters

- **marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .
- **default_only** (*boolean*) – If true , specifies that only the default version of the specified engine or engine and major version combination is to be returned.

describe_cache_parameter_groups (*cache_parameter_group_name=None, max_records=None, marker=None*)

The *DescribeCacheParameterGroups* operation returns a list of cache parameter group descriptions. If a

cache parameter group name is specified, the list will contain only the descriptions for that group.

Parameters

- **cache_parameter_group_name** (*string*) – The name of a specific cache parameter group to return details for.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_cache_parameters (*cache_parameter_group_name*, *source=None*,
max_records=None, *marker=None*)

The DescribeCacheParameters operation returns the detailed parameter list for a particular cache parameter group.

Parameters

- **cache_parameter_group_name** (*string*) – The name of a specific cache parameter group to return details for.
- **source** (*string*) – The parameter types to return.

Valid values: *user* | *system* | *engine-default*

Parameters max_records (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_cache_security_groups (*cache_security_group_name=None*, *max_records=None*,
marker=None)

The DescribeCacheSecurityGroups operation returns a list of cache security group descriptions. If a cache security group name is specified, the list will contain only the description of that group.

Parameters

- **cache_security_group_name** (*string*) – The name of the cache security group to return details for.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_cache_subnet_groups (*cache_subnet_group_name=None*, *max_records=None*,
marker=None)

The DescribeCacheSubnetGroups operation returns a list of cache subnet group descriptions. If a subnet group name is specified, the list will contain only the description of that group.

Parameters

- **cache_subnet_group_name** (*string*) – The name of the cache subnet group to return details for.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_engine_default_parameters (*cache_parameter_group_family*,
max_records=None, marker=None)

The DescribeEngineDefaultParameters operation returns the default engine and system parameter information for the specified cache engine.

Parameters

- **cache_parameter_group_family** (*string*) – The name of the cache parameter group family. Valid values are: *memcached1.4* | *redis2.6*
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_events (*source_identifier=None, source_type=None, start_time=None, end_time=None*,
duration=None, max_records=None, marker=None)

The DescribeEvents operation returns events related to cache clusters, cache security groups, and cache parameter groups. You can obtain events specific to a particular cache cluster, cache security group, or cache parameter group by providing the name as a parameter.

By default, only the events occurring within the last hour are returned; however, you can retrieve up to 14 days' worth of events if necessary.

Parameters

- **source_identifier** (*string*) – The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.
- **source_type** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.

Valid values are: *cache-cluster* | *cache-parameter-group* | *'cache-security-group'* | *cache-subnet-group*

Parameters

- **start_time** (*timestamp*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format.
- **end_time** (*timestamp*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format.
- **duration** (*integer*) – The number of minutes' worth of events to retrieve.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_replication_groups (*replication_group_id=None, marker=None, max_records=None*)

The DescribeReplicationGroups operation returns information about a particular replication group. If no identifier is specified, DescribeReplicationGroups returns information about all replication groups.

Parameters replication_group_id (*string*) – The identifier for the replication group to be described. This parameter is not case sensitive.

If you do not specify this parameter, information about all replication groups is returned.

Parameters max_records (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_reserved_cache_nodes (*reserved_cache_node_id=None, reserved_cache_nodes_offering_id=None, cache_node_type=None, duration=None, product_description=None, offering_type=None, max_records=None, marker=None*)

The DescribeReservedCacheNodes operation returns information about reserved cache nodes for this account, or about a specified reserved cache node.

Parameters

- **reserved_cache_node_id** (*string*) – The reserved cache node identifier filter value. Use this parameter to show only the reservation that matches the specified reservation ID.
- **reserved_cache_nodes_offering_id** (*string*) – The offering identifier filter value. Use this parameter to show only purchased reservations matching the specified offering identifier.
- **cache_node_type** (*string*) – The cache node type filter value. Use this parameter to show only those reservations matching the specified cache node type.

- **duration** (*string*) – The duration filter value, specified in years or seconds. Use this parameter to show only reservations for this duration.

Valid Values: 1 | 3 | 31536000 | 94608000

Parameters

- **product_description** (*string*) – The product description filter value. Use this parameter to show only those reservations matching the specified product description.
- **offering_type** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid values: “Light Utilization” | “Medium Utilization” | “Heavy Utilization” ‘

Parameters max_records (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters marker (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords* .

describe_reserved_cache_nodes_offerings (*reserved_cache_nodes_offering_id=None, cache_node_type=None, duration=None, product_description=None, offering_type=None, max_records=None, marker=None*)

The DescribeReservedCacheNodesOfferings operation lists available reserved cache node offerings.

Parameters reserved_cache_nodes_offering_id (*string*) – The offering identifier filter value. Use this parameter to show only the available offering that matches the specified reservation identifier.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

Parameters

- **cache_node_type** (*string*) – The cache node type filter value. Use this parameter to show only the available offerings matching the specified cache node type.
- **duration** (*string*) – Duration filter value, specified in years or seconds. Use this parameter to show only reservations for a given duration.

Valid Values: 1 | 3 | 31536000 | 94608000

Parameters

- **product_description** (*string*) – The product description filter value. Use this parameter to show only the available offerings matching the specified product description.
- **offering_type** (*string*) – The offering type filter value. Use this parameter to show only the available offerings matching the specified offering type.

Valid Values: “Light Utilization” | “Medium Utilization” | “Heavy Utilization” ‘

Parameters max_records (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20; maximum 100.

Parameters **marker** (*string*) – An optional marker returned from a prior request. Use this marker for pagination of results from this operation. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by MaxRecords .

modify_cache_cluster (*cache_cluster_id*, *num_cache_nodes=None*, *cache_node_ids_to_remove=None*, *cache_security_group_names=None*, *security_group_ids=None*, *preferred_maintenance_window=None*, *notification_topic_arn=None*, *cache_parameter_group_name=None*, *notification_topic_status=None*, *apply_immediately=None*, *engine_version=None*, *auto_minor_version_upgrade=None*)

The ModifyCacheCluster operation modifies the settings for a cache cluster. You can use this operation to change one or more cluster configuration parameters by specifying the parameters and the new values.

Parameters

- **cache_cluster_id** (*string*) – The cache cluster identifier. This value is stored as a lowercase string.
- **num_cache_nodes** (*integer*) – The number of cache nodes that the cache cluster should have. If the value for NumCacheNodes is greater than the existing number of cache nodes, then more nodes will be added. If the value is less than the existing number of cache nodes, then cache nodes will be removed.

If you are removing cache nodes, you must use the CacheNodeIdsToRemove parameter to provide the IDs of the specific cache nodes to be removed.

Parameters

- **cache_node_ids_to_remove** (*list*) – A list of cache node IDs to be removed. A node ID is a numeric identifier (0001, 0002, etc.). This parameter is only valid when NumCacheNodes is less than the existing number of cache nodes. The number of cache node IDs supplied in this parameter must match the difference between the existing number of cache nodes in the cluster and the value of NumCacheNodes in the request.
- **cache_security_group_names** (*list*) – A list of cache security group names to authorize on this cache cluster. This change is asynchronously applied as soon as possible.

This parameter can be used only with clusters that are created outside of an Amazon Virtual Private Cloud (VPC).

Constraints: Must contain no more than 255 alphanumeric characters. Must not be “Default”.

Parameters **security_group_ids** (*list*) – Specifies the VPC Security Groups associated with the cache cluster.

This parameter can be used only with clusters that are created in an Amazon Virtual Private Cloud (VPC).

Parameters

- **preferred_maintenance_window** (*string*) – The weekly time range (in UTC) during which system maintenance can occur. Note that system maintenance may result in an outage. This change is made immediately. If you are moving this window to the current time, there must be at least 120 minutes between the current time and end of the window to ensure that pending changes are applied.
- **notification_topic_arn** (*string*) –

The Amazon Resource Name (ARN) of the SNS topic to which notifications will be sent.

The SNS topic owner must be same as the cache cluster owner.

Parameters

- **cache_parameter_group_name** (*string*) – The name of the cache parameter group to apply to this cache cluster. This change is asynchronously applied as soon as possible for parameters when the `ApplyImmediately` parameter is specified as `true` for this request.
- **notification_topic_status** (*string*) – The status of the Amazon SNS notification topic. Notifications are sent only if the status is `active`.

Valid values: `active` | `inactive`

Parameters **apply_immediately** (*boolean*) – If `True`, this parameter causes the modifications in this request and any pending modifications to be applied, asynchronously and as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the cache cluster.

If `False`, then changes to the cache cluster are applied on the next maintenance reboot, or the next failure reboot, whichever occurs first.

Valid values: `True` | `False`

Default: `False`

Parameters

- **engine_version** (*string*) – The upgraded version of the cache engine to be run on the cache cluster nodes.
- **auto_minor_version_upgrade** (*boolean*) – If `True`, then minor engine upgrades will be applied automatically to the cache cluster during the maintenance window.

Valid values: `True` | `False`

Default: `True`

modify_cache_parameter_group (*cache_parameter_group_name, parameter_name_values*)

The `ModifyCacheParameterGroup` operation modifies the parameters of a cache parameter group. You can modify up to 20 parameters in a single request by submitting a list parameter name and value pairs.

Parameters

- **cache_parameter_group_name** (*string*) – The name of the cache parameter group to modify.
- **parameter_name_values** (*list*) – An array of parameter names and values for the parameter update. You must supply at least one parameter name and value; subsequent arguments are optional. A maximum of 20 parameters may be modified per request.

modify_cache_subnet_group (*cache_subnet_group_name, cache_subnet_group_description=None, subnet_ids=None*)

The `ModifyCacheSubnetGroup` operation modifies an existing cache subnet group.

Parameters **cache_subnet_group_name** (*string*) – The name for the cache subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

Example: `mysubnetgroup`

Parameters

- **cache_subnet_group_description** (*string*) – A description for the cache subnet group.

- **subnet_ids** (*list*) – The EC2 subnet IDs for the cache subnet group.

modify_replication_group (*replication_group_id*, *replication_group_description=None*, *cache_security_group_names=None*, *security_group_ids=None*, *preferred_maintenance_window=None*, *notification_topic_arn=None*, *cache_parameter_group_name=None*, *notification_topic_status=None*, *apply_immediately=None*, *engine_version=None*, *auto_minor_version_upgrade=None*, *primary_cluster_id=None*)

The ModifyReplicationGroup operation modifies the settings for a replication group.

Parameters

- **replication_group_id** (*string*) – The identifier of the replication group to modify.
- **replication_group_description** (*string*) – A description for the replication group. Maximum length is 255 characters.
- **cache_security_group_names** (*list*) – A list of cache security group names to authorize for the clusters in this replication group. This change is asynchronously applied as soon as possible.

This parameter can be used only with replication groups containing cache clusters running outside of an Amazon Virtual Private Cloud (VPC).

Constraints: Must contain no more than 255 alphanumeric characters. Must not be “Default”.

Parameters **security_group_ids** (*list*) – Specifies the VPC Security Groups associated with the cache clusters in the replication group.

This parameter can be used only with replication groups containing cache clusters running in an Amazon Virtual Private Cloud (VPC).

Parameters

- **preferred_maintenance_window** (*string*) – The weekly time range (in UTC) during which replication group system maintenance can occur. Note that system maintenance may result in an outage. This change is made immediately. If you are moving this window to the current time, there must be at least 120 minutes between the current time and end of the window to ensure that pending changes are applied.
- **notification_topic_arn** (*string*) –

The Amazon Resource Name (ARN) of the SNS topic to which notifications will be sent.

The SNS topic owner must be same as the replication group owner.

Parameters

- **cache_parameter_group_name** (*string*) – The name of the cache parameter group to apply to all of the cache nodes in this replication group. This change is asynchronously applied as soon as possible for parameters when the ApplyImmediately parameter is specified as true for this request.
- **notification_topic_status** (*string*) – The status of the Amazon SNS notification topic for the replication group. Notifications are sent only if the status is active .

Valid values: *active* | *inactive*

Parameters **apply_immediately** (*boolean*) – If *True*, this parameter causes the modifications in this request and any pending modifications to be applied, asynchronously and as soon as possible, regardless of the PreferredMaintenanceWindow setting for the replication group.

If *False*, then changes to the nodes in the replication group are applied on the next maintenance reboot, or the next failure reboot, whichever occurs first.

Valid values: *True* | *False*

Default: *False*

Parameters

- **engine_version** (*string*) – The upgraded version of the cache engine to be run on the nodes in the replication group..
- **auto_minor_version_upgrade** (*boolean*) – Determines whether minor engine upgrades will be applied automatically to all of the cache nodes in the replication group during the maintenance window. A value of *True* allows these upgrades to occur; *False* disables automatic upgrades.
- **primary_cluster_id** (*string*) – If this parameter is specified, ElastiCache will promote each of the nodes in the specified cache cluster to the primary role. The nodes of all other clusters in the replication group will be read replicas.

purchase_reserved_cache_nodes_offering (*reserved_cache_nodes_offering_id*,
reserved_cache_node_id=None,
cache_node_count=None)

The PurchaseReservedCacheNodesOffering operation allows you to purchase a reserved cache node offering.

Parameters **reserved_cache_nodes_offering_id** (*string*) – The ID of the reserved cache node offering to purchase.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

Parameters **reserved_cache_node_id** (*string*) – A customer-specified identifier to track this reservation.

Example: myreservationID

Parameters **cache_node_count** (*integer*) – The number of cache node instances to reserve.

Default: *1*

reboot_cache_cluster (*cache_cluster_id*, *cache_node_ids_to_reboot*)

The RebootCacheCluster operation reboots some, or all, of the cache cluster nodes within a provisioned cache cluster. This API will apply any modified cache parameter groups to the cache cluster. The reboot action takes place as soon as possible, and results in a momentary outage to the cache cluster. During the reboot, the cache cluster status is set to REBOOTING.

The reboot causes the contents of the cache (for each cache cluster node being rebooted) to be lost.

When the reboot is complete, a cache cluster event is created.

Parameters

- **cache_cluster_id** (*string*) – The cache cluster identifier. This parameter is stored as a lowercase string.
- **cache_node_ids_to_reboot** (*list*) – A list of cache cluster node IDs to reboot. A node ID is a numeric identifier (0001, 0002, etc.). To reboot an entire cache cluster, specify all of the cache cluster node IDs.

reset_cache_parameter_group (*cache_parameter_group_name*, *parameter_name_values*, *reset_all_parameters=None*)

The ResetCacheParameterGroup operation modifies the parameters of a cache parameter group to the engine or system default value. You can reset specific parameters by submitting a list of parameter names. To reset the entire cache parameter group, specify the ResetAllParameters and CacheParameterGroupName parameters.

Parameters

- **cache_parameter_group_name** (*string*) – The name of the cache parameter group to reset.
- **reset_all_parameters** (*boolean*) – If true , all parameters in the cache parameter group will be reset to default values. If false , no such action occurs.

Valid values: *True* | *False*

Parameters **parameter_name_values** (*list*) – An array of parameter names to be reset. If you are not resetting the entire cache parameter group, you must specify at least one parameter name.

revoke_cache_security_group_ingress (*cache_security_group_name*,
ec2_security_group_name,
ec2_security_group_owner_id)

The RevokeCacheSecurityGroupIngress operation revokes ingress from a cache security group. Use this operation to disallow access from an Amazon EC2 security group that had been previously authorized.

Parameters

- **cache_security_group_name** (*string*) – The name of the cache security group to revoke ingress from.
- **ec2_security_group_name** (*string*) – The name of the Amazon EC2 security group to revoke access from.
- **ec2_security_group_owner_id** (*string*) – The AWS account number of the Amazon EC2 security group owner. Note that this is not the same thing as an AWS access key ID - you must provide a valid AWS account number for this parameter.

4.130 Elastic Transcoder

4.130.1 boto.elastictranscoder

`boto.elastictranscoder.connect_to_region` (*region_name*, ***kw_params*)

`boto.elastictranscoder.regions` ()

Get all available regions for the AWS Elastic Transcoder service.

Return type *list*

Returns A list of `boto.regioninfo.RegionInfo`

4.130.2 boto.elastictranscoder.layer1

class `boto.elastictranscoder.layer1.ElasticTranscoderConnection` (***kwargs*)

AWS Elastic Transcoder Service The AWS Elastic Transcoder Service.

APIVersion = '2012-09-25'

DefaultRegionEndpoint = 'elastictranscoder.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

cancel_job (*id=None*)

The CancelJob operation cancels an unfinished job. You can only cancel a job that has a status of *Submitted*. To prevent a pipeline from starting to process a job while you're getting the job identifier, use `UpdatePipelineStatus` to temporarily pause the pipeline.

Parameters `id` (*string*) – The identifier of the job that you want to cancel.

To get a list of the jobs (including their *jobId*) that have a status of *Submitted*, use the `ListJobsByStatus` API action.

create_job (*pipeline_id=None, input_name=None, output=None, outputs=None, output_key_prefix=None, playlists=None*)

When you create a job, Elastic Transcoder returns JSON data that includes the values that you specified plus information about the job that is created.

If you have specified more than one output for your jobs (for example, one output for the Kindle Fire and another output for the Apple iPhone 4s), you currently must use the Elastic Transcoder API to list the jobs (as opposed to the AWS Console).

Parameters

- **pipeline_id** (*string*) – The *Id* of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.
- **input_name** (*dict*) – A section of the request body that provides information about the file that is being transcoded.
- **output** (*dict*) – The *CreateJobOutput* structure.
- **outputs** (*list*) – A section of the request body that provides information about the transcoded (target) files. We recommend that you use the *Outputs* syntax instead of the *Output* syntax.
- **output_key_prefix** (*string*) – The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists.
- **playlists** (*list*) – If you specify a preset in *PresetId* for which the value of *Container* is *ts* (MPEG-TS), *Playlists* contains information about the master playlists that you want Elastic Transcoder to create.

We recommend that you create only one master playlist. The maximum number of master playlists in a job is 30.

create_pipeline (*name=None, input_bucket=None, output_bucket=None, role=None, notifications=None, content_config=None, thumbnail_config=None*)

The `CreatePipeline` operation creates a pipeline with settings that you specify.

Parameters `name` (*string*) – The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters.

Parameters

- **input_bucket** (*string*) – The Amazon S3 bucket in which you saved the media files that you want to transcode.
- **output_bucket** (*string*) – The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. (Use this, or use `ContentConfig:Bucket` plus `ThumbnailConfig:Bucket`.)

Specify this value when all of the following are true:

- **You want to save transcoded files, thumbnails (if any), and playlists** (if any) together in one bucket.
- **You do not want to specify the users or groups who have access to the** transcoded files, thumbnails, and playlists.

- **You do not want to specify the permissions that Elastic Transcoder** grants to the files. When Elastic Transcoder saves files in *OutputBucket*, it grants full control over the files only to the AWS account that owns the role that is specified by *Role*.
- **You want to associate the transcoded files and thumbnails with the** Amazon S3 Standard storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit *OutputBucket* and specify values for *ContentConfig* and *ThumbnailConfig* instead.

Parameters

- **role** (*string*) – The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to create the pipeline.
- **notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing: The topic ARN for the Amazon Simple Notification** Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see *Create a Topic* in the Amazon Simple Notification Service Developer Guide.
- **Completed: The topic ARN for the Amazon SNS topic that you want** to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Warning: The topic ARN for the Amazon SNS topic that you want to** notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Error: The topic ARN for the Amazon SNS topic that you want to** notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Parameters **content_config** (*dict*) –

The optional *ContentConfig* object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for *ContentConfig*, you must also specify values for *ThumbnailConfig*.

If you specify values for *ContentConfig* and *ThumbnailConfig*, omit the *OutputBucket* object.

- **Bucket: The Amazon S3 bucket in which you want Elastic Transcoder** to save transcoded files and playlists.
- **Permissions (Optional): The Permissions object specifies which** users you want to have access to transcoded files and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **Grantee Type: Specify the type of value that appears in the**

Grantee object:

- **Canonical: The value in the *Grantee* object is either the** canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see *Access Control List (ACL) Overview* in the Amazon Simple Storage Service Developer Guide. For more information about using CloudFront origin access identities to require that users use

CloudFront URLs instead of Amazon S3 URLs, see [Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content](#). A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the *Grantee* object is the registered email address of an AWS account.
- **Group:** The value in the *Grantee* object is one of the following predefined Amazon S3 groups: *AllUsers*, *AuthenticatedUsers*, or *LogDelivery*.
- **Grantee:** The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group
- **Access:** The permission that you want to give to the AWS user that you specified in *Grantee*. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists and video files. Valid values include:
 - **READ:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - **READ_ACP:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - **WRITE_ACP:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - **FULL_CONTROL:** The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass:** The Amazon S3 storage class, *Standard* or *ReducedRedundancy*, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

Parameters `thumbnail_config` (*dict*) –

The *ThumbnailConfig* object specifies several values, including the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for *ContentConfig*, you must also specify values for *ThumbnailConfig* even if you don't want to create thumbnails.

If you specify values for *ContentConfig* and *ThumbnailConfig*, omit the *OutputBucket* object.

- **Bucket:** The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions (Optional):** The *Permissions* object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **GranteeType:** Specify the type of value that appears in the

Grantee object:

- **Canonical:** The value in the *Grantee* object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. A canonical user ID is not the same as an AWS account number.
- **Email:** The value in the *Grantee* object is the registered email address of an AWS account.

- **Group:** The value in the *Grantee* object is one of the following predefined Amazon S3 groups: *AllUsers*, *AuthenticatedUsers*, or *LogDelivery*.
- **Grantee:** The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.
- **Access:** The permission that you want to give to the AWS user that you specified in *Grantee*. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:
 - **READ:** The grantee can read the thumbnails and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - **READ_ACP:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
 - **WRITE_ACP:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
 - **FULL_CONTROL:** The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass:** The Amazon S3 storage class, *Standard* or *ReducedRedundancy*, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

create_preset (*name=None, description=None, container=None, video=None, audio=None, thumbnails=None*)

The `CreatePreset` operation creates a preset with settings that you specify. Elastic Transcoder checks the `CreatePreset` settings to ensure that they meet Elastic Transcoder requirements and to determine whether they comply with H.264 standards. If your settings are not valid for Elastic Transcoder, Elastic Transcoder returns an HTTP 400 response (*ValidationException*) and does not create the preset. If the settings are valid for Elastic Transcoder but aren't strictly compliant with the H.264 standard, Elastic Transcoder creates the preset and returns a warning message in the response. This helps you determine whether your settings comply with the H.264 standard while giving you greater flexibility with respect to the video that Elastic Transcoder produces. Elastic Transcoder uses the H.264 video-compression format. For more information, see the International Telecommunication Union publication Recommendation ITU-T H.264: Advanced video coding for generic audiovisual services .

Parameters

- **name** (*string*) – The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.
- **description** (*string*) – A description of the preset.
- **container** (*string*) – The container type for the output file. Valid values include *mp3*, *mp4*, *ogg*, *ts*, and *webm*.
- **video** (*dict*) – A section of the request body that specifies the video parameters.
- **audio** (*dict*) – A section of the request body that specifies the audio parameters.
- **thumbnails** (*dict*) – A section of the request body that specifies the thumbnail parameters, if any.

delete_pipeline (*id=None*)

The `DeletePipeline` operation removes a pipeline.

You can only delete a pipeline that has never been used or that is not currently in use (doesn't contain any active jobs). If the pipeline is currently in use, *DeletePipeline* returns an error.

Parameters *id* (*string*) – The identifier of the pipeline that you want to delete.

delete_preset (*id=None*)

The DeletePreset operation removes a preset that you've added in an AWS region.

You can't delete the default presets that are included with Elastic Transcoder.

Parameters *id* (*string*) – The identifier of the preset for which you want to get detailed information.

list_jobs_by_pipeline (*pipeline_id=None, ascending=None, page_token=None*)

The ListJobsByPipeline operation gets a list of the jobs currently in a pipeline.

Elastic Transcoder returns all of the jobs currently in the specified pipeline. The response body contains one element for each job that satisfies the search criteria.

Parameters

- **pipeline_id** (*string*) – The ID of the pipeline for which you want to get job information.
- **ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter *True*. To list jobs in reverse chronological order, enter *False*.
- **page_token** (*string*) – When Elastic Transcoder returns more than one page of results, use *pageToken* in subsequent *GET* requests to get each successive page of results.

list_jobs_by_status (*status=None, ascending=None, page_token=None*)

The ListJobsByStatus operation gets a list of jobs that have a specified status. The response body contains one element for each job that satisfies the search criteria.

Parameters

- **status** (*string*) – To get information about all of the jobs associated with the current AWS account that have a given status, specify the following status: *Submitted*, *Progressing*, *Complete*, *Canceled*, or *Error*.
- **ascending** (*string*) – To list jobs in chronological order by the date and time that they were submitted, enter *True*. To list jobs in reverse chronological order, enter *False*.
- **page_token** (*string*) – When Elastic Transcoder returns more than one page of results, use *pageToken* in subsequent *GET* requests to get each successive page of results.

list_pipelines (*ascending=None, page_token=None*)

The ListPipelines operation gets a list of the pipelines associated with the current AWS account.

Parameters

- **ascending** (*string*) – To list pipelines in chronological order by the date and time that they were created, enter *True*. To list pipelines in reverse chronological order, enter *False*.
- **page_token** (*string*) – When Elastic Transcoder returns more than one page of results, use *pageToken* in subsequent *GET* requests to get each successive page of results.

list_presets (*ascending=None, page_token=None*)

The ListPresets operation gets a list of the default presets included with Elastic Transcoder and the presets that you've added in an AWS region.

Parameters

- **ascending** (*string*) – To list presets in chronological order by the date and time that they were created, enter *True*. To list presets in reverse chronological order, enter *False*.
- **page_token** (*string*) – When Elastic Transcoder returns more than one page of results, use *pageToken* in subsequent *GET* requests to get each successive page of results.

make_request (*verb, resource, headers=None, data="", expected_status=None, params=None*)

Makes a request to the server, with stock multiple-retry logic.

read_job (*id=None*)

The ReadJob operation returns detailed information about a job.

Parameters **id** (*string*) – The identifier of the job for which you want to get detailed information.

read_pipeline (*id=None*)

The ReadPipeline operation gets detailed information about a pipeline.

Parameters **id** (*string*) – The identifier of the pipeline to read.

read_preset (*id=None*)

The ReadPreset operation gets detailed information about a preset.

Parameters **id** (*string*) – The identifier of the preset for which you want to get detailed information.

test_role (*role=None, input_bucket=None, output_bucket=None, topics=None*)

The TestRole operation tests the IAM role used to create the pipeline.

The *TestRole* action lets you determine whether the IAM role you are using has sufficient permissions to let Elastic Transcoder perform tasks associated with the transcoding process. The action attempts to assume the specified IAM role, checks read access to the input and output buckets, and tries to send a test notification to Amazon SNS topics that you specify.

Parameters

- **role** (*string*) – The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to test.
- **input_bucket** (*string*) – The Amazon S3 bucket that contains media files to be transcoded. The action attempts to read from this bucket.
- **output_bucket** (*string*) – The Amazon S3 bucket that Elastic Transcoder will write transcoded media files to. The action attempts to read from this bucket.
- **topics** (*list*) – The ARNs of one or more Amazon Simple Notification Service (Amazon SNS) topics that you want the action to send a test notification to.

update_pipeline (*id, name=None, input_bucket=None, role=None, notifications=None, content_config=None, thumbnail_config=None*)

Use the *UpdatePipeline* operation to update settings for a pipeline. When you change pipeline settings, your changes take effect immediately. Jobs that you have already submitted and that Elastic Transcoder has not started to process are affected in addition to jobs that you submit after you change settings.

Parameters

- **id** (*string*) – The ID of the pipeline that you want to update.
- **name** (*string*) – The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

Parameters

- **input_bucket** (*string*) – The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.
- **role** (*string*) – The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.
- **notifications** (*dict*) –

The Amazon Simple Notification Service (Amazon SNS) topic or topics to notify in order to report job status.

To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Parameters **content_config** (*dict*) –

The optional *ContentConfig* object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for *ContentConfig*, you must also specify values for *ThumbnailConfig*.

If you specify values for *ContentConfig* and *ThumbnailConfig*, omit the *OutputBucket* object.

- **Bucket:** The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- **Permissions (Optional):** The Permissions object specifies which users you want to have access to transcoded files and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **Grantee Type:** Specify the type of value that appears in the

Grantee object:

- **Canonical:** The value in the *Grantee* object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content. A canonical user ID is not the same as an AWS account number.
- **Email:** The value in the *Grantee* object is the registered email address of an AWS account.
- **Group:** The value in the *Grantee* object is one of the following predefined Amazon S3 groups: *AllUsers*, *AuthenticatedUsers*, or *LogDelivery*.
- **Grantee:** The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group
- **Access:** The permission that you want to give to the AWS user that you specified in *Grantee*. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists and video files. Valid values include:
 - **READ:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.

- **READ_ACP:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **WRITE_ACP:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **FULL_CONTROL:** The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **StorageClass:** The Amazon S3 storage class, **Standard** or **ReducedRedundancy**, that you want Elastic Transcoder to assign to the video files and playlists that it stores in your Amazon S3 bucket.

Parameters `thumbnail_config` (*dict*) –

The `ThumbnailConfig` object specifies several values, including the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

- **Bucket:** The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- **Permissions (Optional):** The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.
- **GranteeType:** Specify the type of value that appears in the

Grantee object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. A canonical user ID is not the same as an AWS account number.
- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.
- **Grantee:** The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group.
- **Access:** The permission that you want to give to the AWS user that you specified in `Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:
 - **READ:** The grantee can read the thumbnails and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
 - **READ_ACP:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
 - **WRITE_ACP:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

– **FULL_CONTROL**: The grantee has **READ**, **READ_ACP**, and **WRITE_ACP** permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

- **StorageClass**: The Amazon S3 storage class, **Standard** or **ReducedRedundancy**, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

update_pipeline_notifications (*id=None, notifications=None*)

With the UpdatePipelineNotifications operation, you can update Amazon Simple Notification Service (Amazon SNS) notifications for a pipeline.

When you update notifications for a pipeline, Elastic Transcoder returns the values that you specified in the request.

Parameters

- **id** (*string*) – The identifier of the pipeline for which you want to change notification settings.
- **notifications** (*dict*) –

The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status.

To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

- **Progressing**: The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process jobs that are added to this pipeline. This is the ARN that Amazon SNS returned when you created the topic.
- **Completed**: The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job. This is the ARN that Amazon SNS returned when you created the topic.
- **Warning**: The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition. This is the ARN that Amazon SNS returned when you created the topic.
- **Error**: The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition. This is the ARN that Amazon SNS returned when you created the topic.

update_pipeline_status (*id=None, status=None*)

The UpdatePipelineStatus operation pauses or reactivates a pipeline, so that the pipeline stops or restarts the processing of jobs.

Changing the pipeline status is useful if you want to cancel one or more jobs. You can't cancel jobs after Elastic Transcoder has started processing them; if you pause the pipeline to which you submitted the jobs, you have more time to get the job IDs for the jobs that you want to cancel, and to send a CancelJob request.

Parameters

- **id** (*string*) – The identifier of the pipeline to update.
- **status** (*string*) –

The desired status of the pipeline:

- *Active*: The pipeline is processing jobs.
- *Paused*: The pipeline is not currently processing jobs.

4.130.3 boto.elastictranscoder.exceptions

```

exception boto.elastictranscoder.exceptions.AccessDeniedException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.IncompatibleVersionException (status,
                                                                    rea-
                                                                    son,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.InternalServiceException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.LimitExceededException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.ResourceInUseException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.ResourceNotFoundException (status,
                                                                    rea-
                                                                    son,
                                                                    body=None,
                                                                    *args)

exception boto.elastictranscoder.exceptions.ValidationException (status, reason,
                                                                    body=None,
                                                                    *args)

```

4.131 ELB Reference

4.131.1 boto.ec2.elb

This module provides an interface to the Elastic Compute Cloud (EC2) load balancing service from AWS.

```

class boto.ec2.elb.ELBConnection (aws_access_key_id=None, aws_secret_access_key=None,
                                     is_secure=True, port=None, proxy=None, proxy_port=None,
                                     proxy_user=None, proxy_pass=None, debug=0,
                                     https_connection_factory=None, region=None, path='/',
                                     security_token=None, validate_certs=True, pro-
                                     file_name=None)

```

Init method to create a new connection to EC2 Load Balancing Service.

Note: The region argument is overridden by the region specified in the boto configuration file.

```

APIVersion = '2012-06-01'

```

```
DefaultRegionEndpoint = 'elasticloadbalancing.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

apply_security_groups_to_lb (*name, security_groups*)

Associates one or more security groups with the load balancer. The provided security groups will override any currently applied security groups.

Parameters

- **name** (*string*) – The name of the Load Balancer
- **security_groups** (*List of strings*) – The name of the security group(s) to add.

Return type List of strings

Returns An updated list of security groups for this Load Balancer.

attach_lb_to_subnets (*name, subnets*)

Attaches load balancer to one or more subnets. Attaching subnets that are already registered with the Load Balancer has no effect.

Parameters

- **name** (*string*) – The name of the Load Balancer
- **subnets** (*List of strings*) – The name of the subnet(s) to add.

Return type List of strings

Returns An updated list of subnets for this Load Balancer.

build_list_params (*params, items, label*)

configure_health_check (*name, health_check*)

Define a health check for the EndPoints.

Parameters

- **name** (*string*) – The mnemonic name associated with the load balancer
- **health_check** (*boto.ec2.elb.healthcheck.HealthCheck*) – A HealthCheck object populated with the desired values.

Return type *boto.ec2.elb.healthcheck.HealthCheck*

Returns The updated *boto.ec2.elb.healthcheck.HealthCheck*

create_app_cookie_stickiness_policy (*name, lb_name, policy_name*)

Generates a stickiness policy with sticky session lifetimes that follow that of an application-generated cookie. This policy can only be associated with HTTP listeners.

This policy is similar to the policy created by `CreateLBCookieStickinessPolicy`, except that the lifetime of the special Elastic Load Balancing cookie follows the lifetime of the application-generated cookie specified in the policy configuration. The load balancer only inserts a new stickiness cookie when the application response includes a new application cookie.

If the application cookie is explicitly removed or expires, the session stops being sticky until a new application cookie is issued.

create_lb_cookie_stickiness_policy (*cookie_expiration_period, lb_name, policy_name*)

Generates a stickiness policy with sticky session lifetimes controlled by the lifetime of the browser (user-agent) or a specified expiration period. This policy can only be associated only with HTTP listeners.

When a load balancer implements this policy, the load balancer uses a special cookie to track the backend server instance for each request. When the load balancer receives a request, it first checks to see if this cookie is present in the request. If so, the load balancer sends the request to the application server specified

in the cookie. If not, the load balancer sends the request to a server that is chosen based on the existing load balancing algorithm.

A cookie is inserted into the response for binding subsequent requests from the same user to that server. The validity of the cookie is based on the cookie expiration time, which is specified in the policy configuration.

None may be passed for `cookie_expiration_period`.

create_lb_policy (*lb_name, policy_name, policy_type, policy_attributes*)

Creates a new policy that contains the necessary attributes depending on the policy type. Policies are settings that are saved for your load balancer and that can be applied to the front-end listener, or the back-end application server.

create_load_balancer (*name, zones, listeners=None, subnets=None, security_groups=None, scheme='internet-facing', complex_listeners=None*)

Create a new load balancer for your account. By default the load balancer will be created in EC2. To create a load balancer inside a VPC, parameter `zones` must be set to `None` and `subnets` must not be `None`. The load balancer will be automatically created under the VPC that contains the subnet(s) specified.

Parameters

- **name** (*string*) – The mnemonic name associated with the new load balancer
- **zones** (*List of strings*) – The names of the availability zone(s) to add.
- **listeners** (*List of tuples*) – Each tuple contains three or four values, (LoadBalancerPortNumber, InstancePortNumber, Protocol, [SSLCertificateId]) where LoadBalancerPortNumber and InstancePortNumber are integer values between 1 and 65535, Protocol is a string containing either ‘TCP’, ‘SSL’, HTTP’, or ‘HTTPS’; SSLCertificateID is the ARN of a AWS IAM certificate, and must be specified when doing HTTPS.
- **subnets** (*list of strings*) – A list of subnet IDs in your VPC to attach to your LoadBalancer.
- **security_groups** (*list of strings*) – The security groups assigned to your LoadBalancer within your VPC.
- **scheme** (*string*) – The type of a LoadBalancer. By default, Elastic Load Balancing creates an internet-facing LoadBalancer with a publicly resolvable DNS name, which resolves to public IP addresses.

Specify the value `internal` for this option to create an internal LoadBalancer with a DNS name that resolves to private IP addresses.

This option is only available for LoadBalancers attached to an Amazon VPC.

- **complex_listeners** (*List of tuples*) – Each tuple contains four or five values, (LoadBalancerPortNumber, InstancePortNumber, Protocol, InstanceProtocol, SSLCertificateId).

Where:

- LoadBalancerPortNumber and InstancePortNumber are integer values between 1 and 65535
- Protocol and InstanceProtocol is a string containing either ‘TCP’, ‘SSL’, ‘HTTP’, or ‘HTTPS’
- SSLCertificateId is the ARN of an SSL certificate loaded into AWS IAM

Return type *boto.ec2.elb.loadbalancer.LoadBalancer*

Returns The newly created `boto.ec2.elb.loadbalancer.LoadBalancer`

create_load_balancer_listeners (*name, listeners=None, complex_listeners=None*)

Creates a Listener (or group of listeners) for an existing Load Balancer

Parameters

- **name** (*string*) – The name of the load balancer to create the listeners for
- **listeners** (*List of tuples*) – Each tuple contains three or four values, (LoadBalancerPortNumber, InstancePortNumber, Protocol, [SSLCertificateId]) where LoadBalancerPortNumber and InstancePortNumber are integer values between 1 and 65535, Protocol is a string containing either ‘TCP’, ‘SSL’, HTTP’, or ‘HTTPS’; SSLCertificateID is the ARN of a AWS IAM certificate, and must be specified when doing HTTPS.
- **complex_listeners** (*List of tuples*) – Each tuple contains four or five values, (LoadBalancerPortNumber, InstancePortNumber, Protocol, InstanceProtocol, SSLCertificateId).

Where:

- LoadBalancerPortNumber and InstancePortNumber are integer values between 1 and 65535
- Protocol and InstanceProtocol is a string containing either ‘TCP’, ‘SSL’, ‘HTTP’, or ‘HTTPS’
- SSLCertificateId is the ARN of an SSL certificate loaded into AWS IAM

Returns The status of the request

delete_lb_policy (*lb_name, policy_name*)

Deletes a policy from the LoadBalancer. The specified policy must not be enabled for any listeners.

delete_load_balancer (*name*)

Delete a Load Balancer from your account.

Parameters **name** (*string*) – The name of the Load Balancer to delete

delete_load_balancer_listeners (*name, ports*)

Deletes a load balancer listener (or group of listeners)

Parameters

- **name** (*string*) – The name of the load balancer to create the listeners for
- **ports** (*List int*) – Each int represents the port on the ELB to be removed

Returns The status of the request

deregister_instances (*load_balancer_name, instances*)

Remove Instances from an existing Load Balancer.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **instances** (*List of strings*) – The instance ID’s of the EC2 instances to remove.

Return type List of strings

Returns An updated list of instances for this Load Balancer.

describe_instance_health (*load_balancer_name, instances=None*)

Get current state of all Instances registered to an Load Balancer.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **instances** (*List of strings*) – The instance ID's of the EC2 instances to return status for. If not provided, the state of all instances will be returned.

Return type List of *boto.ec2.elb.instancestate.InstanceState*

Returns list of state info for instances in this Load Balancer.

detach_lb_from_subnets (*name, subnets*)

Detaches load balancer from one or more subnets.

Parameters

- **name** (*string*) – The name of the Load Balancer
- **subnets** (*List of strings*) – The name of the subnet(s) to detach.

Return type List of strings

Returns An updated list of subnets for this Load Balancer.

disable_availability_zones (*load_balancer_name, zones_to_remove*)

Remove availability zones from an existing Load Balancer. All zones must be in the same region as the Load Balancer. Removing zones that are not registered with the Load Balancer has no effect. You cannot remove all zones from an Load Balancer.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **zones** (*List of strings*) – The name of the zone(s) to remove.

Return type List of strings

Returns An updated list of zones for this Load Balancer.

enable_availability_zones (*load_balancer_name, zones_to_add*)

Add availability zones to an existing Load Balancer All zones must be in the same region as the Load Balancer Adding zones that are already registered with the Load Balancer has no effect.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **zones** (*List of strings*) – The name of the zone(s) to add.

Return type List of strings

Returns An updated list of zones for this Load Balancer.

get_all_lb_attributes (*load_balancer_name*)

Gets all Attributes of a Load Balancer

Parameters **load_balancer_name** (*string*) – The name of the Load Balancer

Return type *boto.ec2.elb.attribute.LbAttributes*

Returns The attribute object of the ELB.

get_all_load_balancers (*load_balancer_names=None, marker=None*)

Retrieve all load balancers associated with your account.

Parameters

- **load_balancer_names** (*list*) – An optional list of load balancer names.

- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.

Return type *boto.resultset.ResultSet*

Returns A *ResultSet* containing instances of *boto.ec2.elb.loadbalancer.LoadBalancer*

get_lb_attribute (*load_balancer_name, attribute*)

Gets an attribute of a Load Balancer

This will make an EC2 call for each method call.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **attribute** (*string*) – The attribute you wish to see.
 - *accessLog* - *AccessLogAttribute* instance
 - *crossZoneLoadBalancing* - Boolean
 - *connectingSettings* - *ConnectionSettingAttribute* instance
 - *connectionDraining* - *ConnectionDrainingAttribute* instance

Return type Attribute dependent

Returns The new value for the attribute

modify_lb_attribute (*load_balancer_name, attribute, value*)

Changes an attribute of a Load Balancer

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **attribute** (*string*) – The attribute you wish to change.
 - *crossZoneLoadBalancing* - Boolean (true)
 - *connectingSettings* - *ConnectionSettingAttribute* instance
 - *accessLog* - *AccessLogAttribute* instance
 - *connectionDraining* - *ConnectionDrainingAttribute* instance

Parameters value (*string*) – The new value for the attribute

Return type *bool*

Returns Whether the operation succeeded or not

register_instances (*load_balancer_name, instances*)

Add new Instances to an existing Load Balancer.

Parameters

- **load_balancer_name** (*string*) – The name of the Load Balancer
- **instances** (*List of strings*) – The instance ID’s of the EC2 instances to add.

Return type List of strings

Returns An updated list of instances for this Load Balancer.

set_lb_listener_SSL_certificate (*lb_name, lb_port, ssl_certificate_id*)

Sets the certificate that terminates the specified listener’s SSL connections. The specified certificate replaces any prior certificate that was used on the same LoadBalancer and port.

set_lb_policies_of_backend_server (*lb_name, instance_port, policies*)

Replaces the current set of policies associated with a port on which the back-end server is listening with a new set of policies.

set_lb_policies_of_listener (*lb_name, lb_port, policies*)

Associates, updates, or disables a policy with a listener on the load balancer. Currently only zero (0) or one (1) policy can be associated with a listener.

`boto.ec2.elb.connect_to_region` (*region_name, **kw_params*)

Given a valid region name, return a `boto.ec2.elb.ELBConnection`.

Parameters `region_name` (*str*) – The name of the region to connect to.

Return type `boto.ec2.ELBConnection` or `None`

Returns A connection to the given region, or `None` if an invalid region name is given

`boto.ec2.elb.regions` ()

Get all available regions for the ELB service.

Return type `list`

Returns A list of `boto.RegionInfo` instances

4.131.2 boto.ec2.elb.healthcheck

class `boto.ec2.elb.healthcheck.HealthCheck` (*access_point=None, interval=30, target=None, healthy_threshold=3, timeout=5, unhealthy_threshold=5*)

Represents an EC2 Access Point Health Check. See [Configuring a Health Check](#) for a walkthrough on configuring load balancer health checks.

Variables

- **access_point** (*str*) – The name of the load balancer this health check is associated with.
- **interval** (*int*) – Specifies how many seconds there are between health checks.
- **target** (*str*) – Determines what to check on an instance. See the Amazon [HealthCheck](#) documentation for possible Target values.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

update ()

In the case where you have accessed an existing health check on a load balancer, this method applies this instance's health check values to the load balancer it is attached to.

Note: This method will not do anything if the `access_point` attribute isn't set, as is the case with a newly instantiated `HealthCheck` instance.

4.131.3 boto.ec2.elb.instancestate

class `boto.ec2.elb.instancestate.InstanceState` (*load_balancer=None, description=None, state=None, instance_id=None, reason_code=None*)

Represents the state of an EC2 Load Balancer Instance

Variables

- **load_balancer** (`boto.ec2.elb.loadbalancer.LoadBalancer`) – The load balancer this instance is registered to.

- **description** (*str*) – A description of the instance.
- **instance_id** (*str*) – The EC2 instance ID.
- **reason_code** (*str*) – Provides information about the cause of an OutOfService instance. Specifically, it indicates whether the cause is Elastic Load Balancing or the instance behind the LoadBalancer.
- **state** (*str*) – Specifies the current state of the instance.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.131.4 boto.ec2.elb.listelement

class boto.ec2.elb.listelement.**ListElement**

A `list` subclass that has some additional methods for interacting with Amazon's XML API.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.131.5 boto.ec2.elb.listener

class boto.ec2.elb.listener.**Listener** (*load_balancer=None, load_balancer_port=0, instance_port=0, protocol="", ssl_certificate_id=None, instance_protocol=None*)

Represents an EC2 Load Balancer Listener tuple

endElement (*name, value, connection*)

get_complex_tuple ()

get_tuple ()

startElement (*name, attrs, connection*)

4.131.6 boto.ec2.elb.loadbalancer

class boto.ec2.elb.loadbalancer.**Backend** (*connection=None*)

Backend server description

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.ec2.elb.loadbalancer.**LoadBalancer** (*connection=None, name=None, endpoints=None*)

Represents an EC2 Load Balancer.

Variables

- **connection** (`boto.ec2.elb.ELBConnection`) – The connection this load balancer was instance was instantiated from.
- **listeners** (*list*) – A list of tuples in the form of (<Inbound port>, <Outbound port>, <Protocol>)
- **health_check** (`boto.ec2.elb.healthcheck.HealthCheck`) – The health check policy for this load balancer.

- **policies** (`boto.ec2.elb.policies.Policies`) – Cookie stickiness and other policies.
- **name** (*str*) – The name of the Load Balancer.
- **dns_name** (*str*) – The external DNS name for the balancer.
- **created_time** (*str*) – A date+time string showing when the load balancer was created.
- **instances** (*list*) – A list of `boto.ec2.instanceinfo.InstanceInfo` instances, representing the EC2 instances this load balancer is distributing requests to.
- **availability_zones** (*list*) – The availability zones this balancer covers.
- **canonical_hosted_zone_name** (*str*) – Current CNAME for the balancer.
- **canonical_hosted_zone_name_id** (*str*) – The Route 53 hosted zone ID of this balancer. Needed when creating an Alias record in a Route 53 hosted zone.
- **source_security_group** (`boto.ec2.elb.securitygroup.SecurityGroup`) – The security group that you can use as part of your inbound rules for your load balancer back-end instances to disallow traffic from sources other than your load balancer.
- **subnets** (*list*) – A list of subnets this balancer is on.
- **security_groups** (*list*) – A list of additional security groups that have been applied.
- **vpc_id** (*str*) – The ID of the VPC that this ELB resides within.
- **backends** (*list*) – A list of :py:class:`boto.ec2.elb.loadbalancer.Backend` back-end server descriptions.

apply_security_groups (*security_groups*)

Associates one or more security groups with the load balancer. The provided security groups will override any currently applied security groups.

Parameters **security_groups** (*string or List of strings*) – The name of the security group(s) to add.

attach_subnets (*subnets*)

Attaches load balancer to one or more subnets. Attaching subnets that are already registered with the Load Balancer has no effect.

Parameters **subnets** (*string or List of strings*) – The name of the subnet(s) to add.

configure_health_check (*health_check*)

Configures the health check behavior for the instances behind this load balancer. See [Configuring a Health Check](#) for a walkthrough.

Parameters **health_check** (`boto.ec2.elb.healthcheck.HealthCheck`) – A `HealthCheck` instance that tells the load balancer how to check its instances for health.

create_app_cookie_stickiness_policy (*name, policy_name*)

create_cookie_stickiness_policy (*cookie_expiration_period, policy_name*)

create_lb_policy (*policy_name, policy_type, policy_attribute*)

create_listener (*inPort, outPort=None, proto='tcp'*)

create_listeners (*listeners*)

delete ()

Delete this load balancer.

delete_listener (*inPort*)

delete_listeners (*listeners*)

delete_policy (*policy_name*)

Deletes a policy from the LoadBalancer. The specified policy must not be enabled for any listeners.

deregister_instances (*instances*)

Remove instances from this load balancer. Removing instances that are not registered with the load balancer has no effect.

Parameters instances (*list*) – List of instance IDs (strings) that you'd like to remove from this load balancer.

detach_subnets (*subnets*)

Detaches load balancer from one or more subnets.

Parameters subnets (*string or List of strings*) – The name of the subnet(s) to detach.

disable_cross_zone_load_balancing ()

Turns off CrossZone Load Balancing for this ELB.

Return type `bool`

Returns True if successful, False if not.

disable_zones (*zones*)

Disable availability zones from this Access Point.

Parameters zones (*string or List of strings*) – The name of the zone(s) to add.

enable_cross_zone_load_balancing ()

Turns on CrossZone Load Balancing for this ELB.

Return type `bool`

Returns True if successful, False if not.

enable_zones (*zones*)

Enable availability zones to this Access Point. All zones must be in the same region as the Access Point.

Parameters zones (*string or List of strings*) – The name of the zone(s) to add.

endElement (*name, value, connection*)

get_attributes (*force=False*)

Gets the LbAttributes. The Attributes will be cached.

Parameters force (*bool*) – Ignore cache value and reload.

Return type `boto.ec2.elb.attributes.LbAttributes`

Returns The LbAttributes object

get_instance_health (*instances=None*)

Returns a list of `boto.ec2.elb.instancestate.InstanceState` objects, which show the health of the instances attached to this load balancer.

Return type `list`

Returns A list of `InstanceState` instances, representing the instances attached to this load balancer.

is_cross_zone_load_balancing (*force=False*)

Identifies if the ELB is current configured to do CrossZone Balancing.

Parameters force (*bool*) – Ignore cache value and reload.

Return type `bool`

Returns True if balancing is enabled, False if not.

register_instances (*instances*)

Adds instances to this load balancer. All instances must be in the same region as the load balancer. Adding endpoints that are already registered with the load balancer has no effect.

Parameters *instances* (*list*) – List of instance IDs (strings) that you'd like to add to this load balancer.

set_listener_SSL_certificate (*lb_port*, *ssl_certificate_id*)

set_policies_of_backend_server (*instance_port*, *policies*)

set_policies_of_listener (*lb_port*, *policies*)

startElement (*name*, *attrs*, *connection*)

class boto.ec2.elb.loadbalancer.**LoadBalancerZones** (*connection=None*)

Used to collect the zones for a Load Balancer when `enable_zones` or `disable_zones` are called.

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

4.131.7 boto.ec2.elb.policies

class boto.ec2.elb.policies.**AppCookieStickinessPolicy** (*connection=None*)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

class boto.ec2.elb.policies.**LBCookieStickinessPolicy** (*connection=None*)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

class boto.ec2.elb.policies.**OtherPolicy** (*connection=None*)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

class boto.ec2.elb.policies.**Policies** (*connection=None*)

ELB Policies

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

4.131.8 boto.ec2.elb.securitygroup

class boto.ec2.elb.securitygroup.**SecurityGroup** (*connection=None*)

endElement (*name*, *value*, *connection*)

startElement (*name*, *attrs*, *connection*)

4.131.9 boto.ec2.elb.attributes

```
class boto.ec2.elb.attributes.AccessLogAttribute (connection=None)
    Represents the AccessLog segment of ELB attributes.

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.elb.attributes.ConnectionDrainingAttribute (connection=None)
    Represents the ConnectionDraining segment of ELB attributes.

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.elb.attributes.ConnectionSettingAttribute (connection=None)
    Represents the ConnectionSetting segment of ELB Attributes.

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.elb.attributes.CrossZoneLoadBalancingAttribute (connection=None)
    Represents the CrossZoneLoadBalancing segment of ELB Attributes.

    endElement (name, value, connection)

    startElement (name, attrs, connection)

class boto.ec2.elb.attributes.LbAttributes (connection=None)
    Represents the Attributes of an Elastic Load Balancer.

    endElement (name, value, connection)

    startElement (name, attrs, connection)
```

4.132 EMR

4.132.1 boto.emr

This module provides an interface to the Elastic MapReduce (EMR) service from AWS.

```
boto.emr.connect_to_region (region_name, **kw_params)
```

```
boto.emr.regions ()
```

Get all available regions for the Amazon Elastic MapReduce service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.132.2 boto.emr.connection

Represents a connection to the EMR service


```

class boto.emr.connection.EmrConnection(aws_access_key_id=None,
                                       aws_secret_access_key=None, is_secure=True,
                                       port=None, proxy=None, proxy_port=None,
                                       proxy_user=None, proxy_pass=None, debug=0,
                                       https_connection_factory=None, region=None,
                                       path='/', security_token=None, validate_certs=True,
                                       profile_name=None)

APIVersion = '2009-03-31'
DebuggingArgs = 's3://{region_name}.elasticmapreduce/libs/state-pusher/0.1/fetch'
DebuggingJar = 's3://{region_name}.elasticmapreduce/libs/script-runner/script-runner.jar'
DefaultRegionEndpoint = 'elasticmapreduce.us-east-1.amazonaws.com'
DefaultRegionName = 'us-east-1'

ResponseError
    alias of boto.exception.EmrResponseError

add_instance_groups(jobflow_id, instance_groups)
    Adds instance groups to a running cluster.
    Parameters
        • jobflow_id (str) – The id of the jobflow which will take the new instance groups
        • instance_groups (list (boto.emr.InstanceGroup)) – A list of instance groups to add to the job

add_jobflow_steps(jobflow_id, steps)
    Adds steps to a jobflow
    Parameters
        • jobflow_id (str) – The job flow id
        • steps (list (boto.emr.Step)) – A list of steps to add to the job

add_tags(resource_id, tags)
    Create new metadata tags for the specified resource id.
    Parameters
        • resource_id (str) – The cluster id
        • tags (dict) – A dictionary containing the name/value pairs. If you want to create only a tag name, the value for that tag should be the empty string (e.g. '') or None.

describe_cluster(cluster_id)
    Describes an Elastic MapReduce cluster
    Parameters cluster_id (str) – The cluster id of interest

describe_jobflow(jobflow_id)
    This method is deprecated. We recommend you use list_clusters, describe_cluster, list_steps, list_instance_groups and list_bootstrap_actions instead.
    Describes a single Elastic MapReduce job flow
    Parameters jobflow_id (str) – The job flow id of interest

describe_jobflows(states=None, jobflow_ids=None, created_after=None, created_before=None)
    This method is deprecated. We recommend you use list_clusters, describe_cluster, list_steps,

```

`list_instance_groups` and `list_bootstrap_actions` instead.

Retrieve all the Elastic MapReduce job flows on your account

Parameters

- **states** (*list*) – A list of strings with job flow states wanted
- **jobflow_ids** (*list*) – A list of job flow IDs
- **created_after** (*datetime*) – Bound on job flow creation time
- **created_before** (*datetime*) – Bound on job flow creation time

describe_step (*cluster_id, step_id*)

Describe an Elastic MapReduce step

Parameters

- **cluster_id** (*str*) – The cluster id of interest
- **step_id** (*str*) – The step id of interest

list_bootstrap_actions (*cluster_id, marker=None*)

Get a list of bootstrap actions for an Elastic MapReduce cluster

Parameters

- **cluster_id** (*str*) – The cluster id of interest
- **marker** (*str*) – Pagination marker

list_clusters (*created_after=None, created_before=None, cluster_states=None, marker=None*)

List Elastic MapReduce clusters with optional filtering

Parameters

- **created_after** (*datetime*) – Bound on cluster creation time
- **created_before** (*datetime*) – Bound on cluster creation time
- **cluster_states** (*list*) – Bound on cluster states
- **marker** (*str*) – Pagination marker

list_instance_groups (*cluster_id, marker=None*)

List EC2 instance groups in a cluster

Parameters

- **cluster_id** (*str*) – The cluster id of interest
- **marker** (*str*) – Pagination marker

list_instances (*cluster_id, instance_group_id=None, instance_group_types=None, marker=None*)

List EC2 instances in a cluster

Parameters

- **cluster_id** (*str*) – The cluster id of interest
- **instance_group_id** (*str*) – The EC2 instance group id of interest
- **instance_group_types** (*list*) – Filter by EC2 instance group type
- **marker** (*str*) – Pagination marker

list_steps (*cluster_id, step_states=None, marker=None*)

List cluster steps

Parameters

- **cluster_id** (*str*) – The cluster id of interest

- **step_states** (*list*) – Filter by step states
- **marker** (*str*) – Pagination marker

modify_instance_groups (*instance_group_ids, new_sizes*)

Modify the number of nodes and configuration settings in an instance group.

Parameters

- **instance_group_ids** (*list (str)*) – A list of the ID's of the instance groups to be modified
- **new_sizes** (*list (int)*) – A list of the new sizes for each instance group

remove_tags (*resource_id, tags*)

Remove metadata tags for the specified resource id.

Parameters

- **resource_id** (*str*) – The cluster id
- **tags** (*list*) – A list of tag names to remove.

run_jobflow (*name, log_uri=None, ec2_keyname=None, availability_zone=None, master_instance_type='m1.small', slave_instance_type='m1.small', num_instances=1, action_on_failure='TERMINATE_JOB_FLOW', keep_alive=False, enable_debugging=False, hadoop_version=None, steps=None, bootstrap_actions=[], instance_groups=None, additional_info=None, ami_version=None, api_params=None, visible_to_all_users=None, job_flow_role=None, service_role=None*)

Runs a job flow :type name: str :param name: Name of the job flow

Parameters

- **log_uri** (*str*) – URI of the S3 bucket to place logs
- **ec2_keyname** (*str*) – EC2 key used for the instances
- **availability_zone** (*str*) – EC2 availability zone of the cluster
- **master_instance_type** (*str*) – EC2 instance type of the master
- **slave_instance_type** (*str*) – EC2 instance type of the slave nodes
- **num_instances** (*int*) – Number of instances in the Hadoop cluster
- **action_on_failure** (*str*) – Action to take if a step terminates
- **keep_alive** (*bool*) – Denotes whether the cluster should stay alive upon completion
- **enable_debugging** (*bool*) – Denotes whether AWS console debugging should be enabled.
- **hadoop_version** (*str*) – Version of Hadoop to use. This no longer defaults to '0.20' and now uses the AMI default.
- **steps** (*list (boto.emr.Step)*) – List of steps to add with the job
- **bootstrap_actions** (*list (boto.emr.BootstrapAction)*) – List of bootstrap actions that run before Hadoop starts.
- **instance_groups** (*list (boto.emr.InstanceGroup)*) – Optional list of instance groups to use when creating this job. NB: When provided, this argument supersedes num_instances and master/slave_instance_type.

- **ami_version** (*str*) – Amazon Machine Image (AMI) version to use for instances. Values accepted by EMR are ‘1.0’, ‘2.0’, and ‘latest’; EMR currently defaults to ‘1.0’ if you don’t set ‘ami_version’.
- **additional_info** (*JSON str*) – A JSON string for selecting additional features
- **api_params** (*dict*) – a dictionary of additional parameters to pass directly to the EMR API (so you don’t have to upgrade boto to use new EMR features). You can also delete an API parameter by setting it to None.
- **visible_to_all_users** (*bool*) – Whether the job flow is visible to all IAM users of the AWS account associated with the job flow. If this value is set to `True`, all IAM users of that AWS account can view and (if they have the proper policy permissions set) manage the job flow. If it is set to `False`, only the IAM user that created the job flow can view and manage it.
- **job_flow_role** (*str*) – An IAM role for the job flow. The EC2 instances of the job flow assume this role. The default role is `EMRJobflowDefault`. In order to use the default role, you must have already created it using the CLI.
- **service_role** (*str*) – The IAM role that will be assumed by the Amazon EMR service to access AWS resources on your behalf.

Return type *str*

Returns The jobflow id

set_termination_protection (*jobflow_id, termination_protection_status*)

Set termination protection on specified Elastic MapReduce job flows

Parameters

- **jobflow_ids** (*list or str*) – A list of job flow IDs
- **termination_protection_status** (*bool*) – Termination protection status

set_visible_to_all_users (*jobflow_id, visibility*)

Set whether specified Elastic Map Reduce job flows are visible to all IAM users

Parameters

- **jobflow_ids** (*list or str*) – A list of job flow IDs
- **visibility** (*bool*) – Visibility

terminate_jobflow (*jobflow_id*)

Terminate an Elastic MapReduce job flow

Parameters **jobflow_id** (*str*) – A jobflow id

terminate_jobflows (*jobflow_ids*)

Terminate an Elastic MapReduce job flow

Parameters **jobflow_ids** (*list*) – A list of job flow IDs

4.132.3 boto.emr.step

```
class boto.emr.step.HiveBase (name, **kw)
```

```
    BaseArgs = ['s3n://us-east-1.elasticmapreduce/libs/hive/hive-script', '--base-path', '']
```

```
class boto.emr.step.HiveStep (name, hive_file, hive_versions='latest', hive_args=None)
```

```
    Hive script step
```

```
class boto.emr.step.InstallHiveStep (hive_versions='latest', hive_site=None)
    Install Hive on EMR step
```

```
    InstallHiveName = 'Install Hive'
```

```
class boto.emr.step.InstallPigStep (pig_versions='latest')
    Install pig on emr step
```

```
    InstallPigName = 'Install Pig'
```

```
class boto.emr.step.JarStep (name, jar, main_class=None, action_on_failure='TERMINATE_JOB_FLOW', step_args=None)
```

```
    Custom jar step
```

```
    A elastic mapreduce step that executes a jar
```

Parameters

- **name** (*str*) – The name of the step
- **jar** (*str*) – S3 URI to the Jar file
- **main_class** (*str*) – The class to execute in the jar
- **action_on_failure** (*str*) – An action, defined in the EMR docs to take on failure.
- **step_args** (*list(str)*) – A list of arguments to pass to the step

```
args ()
```

```
    Return type list(str)
```

```
    Returns List of arguments for the step
```

```
jar ()
```

```
    Return type str
```

```
    Returns URI to the jar
```

```
main_class ()
```

```
    Return type str
```

```
    Returns The main class name
```

```
class boto.emr.step.PigBase (name, **kw)
```

```
    BaseArgs = ['s3n://us-east-1.elasticmapreduce/libs/pig/pig-script', '--base-path', 's3n://us-east-1.elasticmapreduce/libs/pig/pig-script']
```

```
class boto.emr.step.PigStep (name, pig_file, pig_versions='latest', pig_args=[])
    Pig script step
```

```
class boto.emr.step.ScriptRunnerStep (name, **kw)
```

```
    ScriptRunnerJar = 's3n://us-east-1.elasticmapreduce/libs/script-runner/script-runner.jar'
```

```
class boto.emr.step.Step
    Jobflow Step base class
```

```
args ()
```

```
    Return type list(str)
```

```
    Returns List of arguments for the step
```

```
jar ()
```

```
    Return type str
```

```
    Returns URI to the jar
```

main_class()

Return type `str`

Returns The main class name

```
class boto.emr.step.StreamingStep(name, mapper, reducer=None, combiner=None,  
                                action_on_failure='TERMINATE_JOB_FLOW',  
                                cache_files=None, cache_archives=None,  
                                step_args=None, input=None, output=None,  
                                jar='/home/hadoop/contrib/streaming/hadoop-  
                                streaming.jar')
```

Hadoop streaming step

A hadoop streaming elastic mapreduce step

Parameters

- **name** (*str*) – The name of the step
- **mapper** (*str*) – The mapper URI
- **reducer** (*str*) – The reducer URI
- **combiner** (*str*) – The combiner URI. Only works for Hadoop 0.20 and later!
- **action_on_failure** (*str*) – An action, defined in the EMR docs to take on failure.
- **cache_files** (*list(str)*) – A list of cache files to be bundled with the job
- **cache_archives** (*list(str)*) – A list of jar archives to be bundled with the job
- **step_args** (*list(str)*) – A list of arguments to pass to the step
- **input** (*str or a list of str*) – The input uri
- **output** (*str*) – The output uri
- **jar** (*str*) – The hadoop streaming jar. This can be either a local path on the master node, or an s3:// URI.

args()

Return type `list(str)`

Returns List of arguments for the step

jar()

Return type `str`

Returns URI to the jar

main_class()

Return type `str`

Returns The main class name

4.132.4 boto.emr.emrobject

This module contains EMR response objects

```
class boto.emr.emrobject.AddInstanceGroupsResponse(connection=None)
```

```
Fields = set(['InstanceGroupIds', 'JobFlowId'])
```

```

class boto.emr.emrobject.Application(connection=None)

    Fields = set(['Args', 'Version', 'Name', 'AdditionalInfo'])
class boto.emr.emrobject.Arg(connection=None)

    endElement(name, value, connection)
class boto.emr.emrobject.BootstrapAction(connection=None)

    Fields = set(['Path', 'Args', 'Name', 'ScriptPath'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.BootstrapActionList(connection=None)

    Fields = set(['Marker'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.Cluster(connection=None)

    Fields = set(['Name', 'ServiceRole', 'TerminationProtected', 'RunningAmiVersion', 'NormalizedInstanceHours'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.ClusterStateChangeReason(connection=None)

    Fields = set(['Message', 'Code'])
class boto.emr.emrobject.ClusterStatus(connection=None)

    Fields = set(['Timeline', 'State', 'StateChangeReason'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.ClusterSummary(connection)

    Fields = set(['NormalizedInstanceHours', 'Id', 'Name'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.ClusterSummaryList(connection)

    Fields = set(['Marker'])
    startElement(name, attrs, connection)
class boto.emr.emrobject.ClusterTimeline(connection=None)

    Fields = set(['ReadyDateTime', 'CreationDateTime', 'EndDateTime'])
class boto.emr.emrobject.Ec2InstanceAttributes(connection=None)

    Fields = set(['Ec2SubnetId', 'IamInstanceProfile', 'Ec2KeyName', 'Ec2AvailabilityZone'])

```

```
class boto.emr.emrobject.EmrObject (connection=None)

    Fields = set([])
    endElement (name, value, connection)
    startElement (name, attrs, connection)
class boto.emr.emrobject.HadoopStep (connection=None)

    Fields = set(['Id', 'ActionOnFailure', 'Name'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.InstanceGroup (connection=None)

    Fields = set(['ReadyDateTime', 'InstanceType', 'InstanceRole', 'EndDateTime', 'InstanceProfileName'])
class boto.emr.emrobject.InstanceGroupInfo (connection=None)

    Fields = set(['RequestedInstanceCount', 'Name', 'InstanceGroupType', 'Id', 'BidPrice', 'SubnetId', 'AvailabilityZone', 'PlacementGroup', 'ElasticMapReduceManaged'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.InstanceGroupList (connection=None)

    Fields = set(['Marker'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.InstanceInfo (connection=None)

    Fields = set(['Ec2InstanceId', 'PublicDnsName', 'PrivateDnsName', 'PublicIpAddress', 'PrivateIpAddress', 'SubnetId', 'AvailabilityZone', 'PlacementGroup', 'ElasticMapReduceManaged'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.InstanceList (connection=None)

    Fields = set(['Marker'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.JobFlow (connection=None)

    Fields = set(['TerminationProtected', 'MasterInstanceId', 'State', 'HadoopVersion', 'LogUri', 'JobFlowRole', 'CreationTime', 'LastModifiedTime', 'JobFlowId', 'JobFlowName', 'JobFlowArn'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.JobFlowStepList (connection=None)

    startElement (name, attrs, connection)
class boto.emr.emrobject.KeyValue (connection=None)

    Fields = set(['Value', 'Key'])
class boto.emr.emrobject.ModifyInstanceGroupsResponse (connection=None)
```



```

    Fields = set(['RequestId'])
class boto.emr.emrobject.RunJobFlowResponse (connection=None)

    Fields = set(['JobFlowId'])
class boto.emr.emrobject.Step (connection=None)

    Fields = set(['Name', 'EndTime', 'Jar', 'ActionOnFailure', 'State', 'MainClass', '
    startElement (name, attrs, connection)
class boto.emr.emrobject.StepConfig (connection=None)

    Fields = set(['MainClass', 'Jar'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.StepId (connection=None)
class boto.emr.emrobject.StepSummary (connection=None)

    Fields = set(['Id', 'Name'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.StepSummaryList (connection=None)

    Fields = set(['Marker'])
    startElement (name, attrs, connection)
class boto.emr.emrobject.SupportedProduct (connection=None)

```

4.133 file

4.133.1 boto.file.bucket

```

class boto.file.bucket.Bucket (name, contained_key)
    Instantiate an anonymous file-based Bucket around a single key.

    delete_key (key_name, headers=None, version_id=None, mfa_token=None)
        Deletes a key from the bucket.
        Parameters
            • key_name (string) – The key name to delete
            • version_id (string) – Unused in this subclass.
            • mfa_token (tuple or list of strings) – Unused in this subclass.

    get_all_keys (headers=None, **params)
        This method returns the single key around which this anonymous Bucket was instantiated.
        Return type SimpleResultSet
        Returns The result from file system listing the keys requested

```

get_key (*key_name*, *headers=None*, *version_id=None*, *key_type=0*)

Check to see if a particular key exists within the bucket. Returns: An instance of a Key object or None

Parameters

- **key_name** (*string*) – The name of the key to retrieve
- **version_id** (*string*) – Unused in this subclass.
- **stream_type** (*integer*) – Type of the Key - Regular File or input/output Stream

Return type *boto.file.key.Key*

Returns A Key object from this bucket.

new_key (*key_name=None*, *key_type=0*)

Creates a new key

Parameters **key_name** (*string*) – The name of the key to create

Return type *boto.file.key.Key*

Returns An instance of the newly created key object

4.133.2 boto.file.simpleresultset

class *boto.file.simpleresultset.SimpleResultSet* (*input_list*)

ResultSet facade built from a simple list, rather than via XML parsing.

4.133.3 boto.file.connection

class *boto.file.connection.FileConnection* (*file_storage_uri*)

get_bucket (*bucket_name*, *validate=True*, *headers=None*)

4.133.4 boto.file.key

class *boto.file.key.Key* (*bucket*, *name*, *fp=None*, *key_type=0*)

KEY_REGULAR_FILE = 0

KEY_STREAM = 3

KEY_STREAM_READABLE = 1

KEY_STREAM_WRITABLE = 2

close ()

Closes fp associated with underlying file. Caller should call this method when done with this class, to avoid using up OS resources (e.g., when iterating over a large number of files).

get_contents_as_string (*headers=None*, *cb=None*, *num_cb=10*, *torrent=False*)

Retrieve file data from the Key, and return contents as a string.

Parameters

- **headers** (*dict*) – ignored in this subclass.
- **cb** (*int*) – ignored in this subclass.

- **num_cb** – ignored in this subclass.
- **num_cb** – ignored in this subclass.
- **torrent** (*bool*) – ignored in this subclass.

Return type string

Returns The contents of the file as a string

get_contents_to_file (*fp*, *headers=None*, *cb=None*, *num_cb=None*, *torrent=False*, *version_id=None*, *res_download_handler=None*, *response_headers=None*)
Copy contents from the current file to the file pointed to by 'fp'.

Parameters

- **fp** (*File-like object*) –
- **headers** (*dict*) – Unused in this subclass.
- **cb** (*int*) – Unused in this subclass.
- **num_cb** – Unused in this subclass.
- **torrent** (*bool*) – Unused in this subclass.
- **res_download_handler** – Unused in this subclass.
- **response_headers** (*dict*) – Unused in this subclass.

get_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*)

Retrieves a file from a Key

Parameters

- **fp** (*file*) – File pointer to put the data into
- **cb** (*int*) – ignored in this subclass.
- **num_cb** – ignored in this subclass.

Param ignored in this subclass.

is_stream ()

set_contents_from_file (*fp*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*, *md5=None*)

Store an object in a file using the name of the Key object as the key in file URI and the contents of the file pointed to by 'fp' as the contents.

Parameters

- **fp** (*file*) – the file whose contents to upload
- **headers** (*dict*) – ignored in this subclass.
- **replace** (*bool*) – If this parameter is False, the method will first check to see if an object exists in the bucket with the same key. If it does, it won't overwrite it. The default value is True which will overwrite the object.
- **cb** (*int*) – ignored in this subclass.
- **num_cb** – ignored in this subclass.
- **policy** (*boto.s3.acl.CannedACLStrings*) – ignored in this subclass.
- **md5** (A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the

second element. This is the same format returned by the compute_md5 method.) – ignored in this subclass.

4.134 fps

4.134.1 boto.fps

4.134.2 boto.fps.connection

```
class boto.fps.connection.FPSConnection(*args, **kw)
```

```
APIVersion = '2010-08-28'
```

```
ResponseError
```

```
    alias of boto.fps.exception.ResponseErrorFactory
```

```
cancel (**kw)
```

```
    FPS Cancel API call
```

```
        Cancels an ongoing transaction and puts it in cancelled state.
```

```
    Required: TransactionId
```

```
cancel_subscription_and_refund (**kw)
```

```
    FPS CancelSubscriptionAndRefund API call
```

```
        Cancels a subscription.
```

```
    Required: SubscriptionId Complex Amounts: RefundAmount Uses CallerReference, defaults to  
    uuid.uuid4()
```

```
cancel_token (**kw)
```

```
    FPS CancelToken API call
```

```
        Cancels any token installed by the calling application on its own account.
```

```
    Required: TokenId
```

```
cbui_url (**kw)
```

```
        Generate a signed URL for the Co-Branded service API given arguments as payload.
```

```
    Required: returnUrl+pipelineName Uses CallerReference, defaults to uuid.uuid4()
```

```
currencycode = 'USD'
```

```
fund_prepaid (**kw)
```

```
    FPS FundPrepaid API call
```

```
        Funds the prepaid balance on the given prepaid instrument.
```

```
    Required: PrepaidInstrumentId+FundingAmount.Value+SenderTokenId+FundingAmount.CurrencyCode  
    Complex Amounts: FundingAmount Uses CallerReference, defaults to uuid.uuid4()
```

```
get_account_activity (**kw)
```

```
    FPS GetAccountActivity API call
```

```
        Returns transactions for a given date range.
```

```
    Required: StartDate
```

```
get_account_balance (*args, **kw)
```

```
    FPS GetAccountBalance API call
```

```
        Returns the account balance for an account in real time.
```

```
get_debt_balance (**kw)
```

```
    FPS GetDebtBalance API call
```

```
        Returns the balance corresponding to the given credit instrument.
```

Required: CreditInstrumentId

get_outstanding_debt_balance (*args, **kw)
FPS GetOutstandingDebtBalance API call

Returns the total outstanding balance for all the credit instruments for the given creditor account.

get_payment_instruction (**kw)
FPS GetPaymentInstruction API call
Gets the payment instruction of a token.
Required: TokenId

get_prepaid_balance (**kw)
FPS GetPrepaidBalance API call
Returns the balance available on the given prepaid instrument.
Required: PrepaidInstrumentId

get_recipient_verification_status (**kw)
FPS GetRecipientVerificationStatus API call
Returns the recipient status.
Required: RecipientTokenId

get_subscription_details (**kw)
FPS GetSubscriptionDetails API call
Returns the details of Subscription for a given subscriptionID.
Required: SubscriptionId

get_token_by_caller (**kw)
FPS GetTokenByCaller API call
Returns the details of a particular token installed by this calling application using the subway co-branded UI.
Required: CallerReference OR TokenId

get_token_usage (**kw)
FPS GetTokenUsage API call
Returns the usage of a token.
Required: TokenId

get_tokens (*args, **kw)
FPS GetTokens API call
Returns a list of tokens installed on the given account.

get_total_prepaid_liability (*args, **kw)
FPS GetTotalPrepaidLiability API call
Returns the total liability held by the given account corresponding to all the prepaid instruments owned by the account.

get_transaction (**kw)
FPS GetTransaction API call
Returns all details of a transaction.
Required: TransactionId

get_transaction_status (**kw)
FPS GetTransactionStatus API call
Gets the latest status of a transaction.
Required: TransactionId

get_transactions_for_subscription (**kw)
FPS GetTransactionsForSubscription API call

Returns the transactions for a given subscriptionID.
Required: SubscriptionId

install_payment_instruction (**kw)
FPS InstallPaymentInstruction API call
Installs a payment instruction for caller.
Required: PaymentInstruction+TokenType Uses CallerReference, defaults to uuid.uuid4()

pay (**kw)
FPS Pay API call
Allows calling applications to move money from a sender to a recipient.
Required: SenderTokenId+TransactionAmount.Value+TransactionAmount.CurrencyCode Complex
Amounts: TransactionAmount Uses CallerReference, defaults to uuid.uuid4()

refund (*args, **kw)
FPS Refund API call
Refunds a previously completed transaction.
Required: TransactionId+RefundAmount.Value+CallerReference+RefundAmount.CurrencyCode Complex
Amounts: RefundAmount

reserve (**kw)
FPS Reserve API call
Reserve API is part of the Reserve and Settle API conjunction that serve the purpose of a pay
where the authorization and settlement have a timing difference.
Required: SenderTokenId+TransactionAmount.Value+TransactionAmount.CurrencyCode Complex
Amounts: TransactionAmount Uses CallerReference, defaults to uuid.uuid4()

settle (*args, **kw)
FPS Settle API call
The Settle API is used in conjunction with the Reserve API and is used to settle previously
reserved transaction.
Required: ReserveTransactionId+TransactionAmount.Value+TransactionAmount.CurrencyCode Complex
Amounts: TransactionAmount

settle_debt (**kw)
FPS SettleDebt API call
Allows a caller to initiate a transaction that atomically transfers money from a sender's pay-
ment instrument to the recipient, while decreasing corresponding debt balance.
Required: CreditInstrumentId+SettlementAmount.Value+SenderTokenId+SettlementAmount.CurrencyCode
Complex Amounts: SettlementAmount Uses CallerReference, defaults to uuid.uuid4()

verify_signature (**kw)
FPS VerifySignature API call
Verify the signature that FPS sent in IPN or callback urls.
Required: UrlEndPoint+HttpParameters

write_off_debt (**kw)
FPS WriteOffDebt API call
Allows a creditor to write off the debt balance accumulated partially or fully at any time.
Required: CreditInstrumentId+AdjustmentAmount.Value+AdjustmentAmount.CurrencyCode Complex
Amounts: AdjustmentAmount Uses CallerReference, defaults to uuid.uuid4()

4.135 Glacier

4.135.1 boto.glacier

`boto.glacier.connect_to_region(region_name, **kw_params)`

`boto.glacier.regions()`

Get all available regions for the Amazon Glacier service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.135.2 boto.glacier.layer1

```
class boto.glacier.layer1.Layer1 (aws_access_key_id=None,  aws_secret_access_key=None,
    account_id='-', is_secure=True, port=None, proxy=None,
    proxy_port=None, proxy_user=None, proxy_pass=None,
    debug=0, https_connection_factory=None,
    path='/', provider='aws', security_token=None,
    suppress_consec_slashes=True, region=None,
    region_name='us-east-1', profile_name=None)
```

Amazon Glacier is a storage solution for “cold data.”

Amazon Glacier is an extremely low-cost storage service that provides secure, durable and easy-to-use storage for data backup and archival. With Amazon Glacier, customers can store their data cost effectively for months, years, or decades. Amazon Glacier also enables customers to offload the administrative burdens of operating and scaling storage to AWS, so they don’t have to worry about capacity planning, hardware provisioning, data replication, hardware failure and recovery, or time-consuming hardware migrations.

Amazon Glacier is a great storage choice when low storage cost is paramount, your data is rarely retrieved, and retrieval latency of several hours is acceptable. If your application requires fast or frequent access to your data, consider using Amazon S3. For more information, go to [‘Amazon Simple Storage Service \(Amazon S3\)’](#).

You can store any kind of data in any format. There is no maximum limit on the total amount of data you can store in Amazon Glacier.

If you are a first-time user of Amazon Glacier, we recommend that you begin by reading the following sections in the Amazon Glacier Developer Guide :

- [‘What is Amazon Glacier’](#) - This section of the Developer Guide describes the underlying data model, the operations it supports, and the AWS SDKs that you can use to interact with the service.
- [‘Getting Started with Amazon Glacier’](#) - The Getting Started section walks you through the process of creating a vault, uploading archives, creating jobs to download archives, retrieving the job output, and deleting archives.

Version = '2012-06-01'

abort_multipart_upload(vault_name, upload_id)

This operation aborts a multipart upload identified by the upload ID.

After the Abort Multipart Upload request succeeds, you cannot upload any more parts to the multipart upload or complete the multipart upload. Aborting a completed upload fails. However, aborting an already-aborted upload will succeed, for a short time. For more information about uploading a part and completing a multipart upload, see `UploadMultipartPart` and `CompleteMultipartUpload`.

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit

permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Working with Archives in Amazon Glacier’](#) and [‘Abort Multipart Upload’](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **upload_id** (*string*) – The upload ID of the multipart upload to delete.

complete_multipart_upload (*vault_name, upload_id, sha256_treehash, archive_size*)

You call this operation to inform Amazon Glacier that all the archive parts have been uploaded and that Amazon Glacier can now assemble the archive from the uploaded parts. After assembling and saving the archive to the vault, Amazon Glacier returns the URI path of the newly created archive resource. Using the URI path, you can then access the archive. After you upload an archive, you should save the archive ID returned to retrieve the archive at a later point. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see [InitiateJob](#).

In the request, you must include the computed SHA256 tree hash of the entire archive you have uploaded. For information about computing a SHA256 tree hash, see [‘Computing Checksums’](#). On the server side, Amazon Glacier also constructs the SHA256 tree hash of the assembled archive. If the values match, Amazon Glacier saves the archive to the vault; otherwise, it returns an error, and the operation fails. The [ListParts](#) operation returns a list of parts uploaded for a specific multipart upload. It includes checksum information for each uploaded part that can be used to debug a bad checksum issue.

Additionally, Amazon Glacier also checks for any missing content ranges when assembling the archive, if missing content ranges are found, Amazon Glacier returns an error and the operation fails.

Complete Multipart Upload is an idempotent operation. After your first successful complete multipart upload, if you call the operation again within a short period, the operation will succeed and return the same archive ID. This is useful in the event you experience a network issue that causes an aborted connection or receive a 500 server error, in which case you can repeat your Complete Multipart Upload request and get the same archive ID without creating duplicate archives. Note, however, that after the multipart upload completes, you cannot call the [List Parts](#) operation and the multipart upload will not appear in [List Multipart Uploads](#) response, even if idempotent complete is possible.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Uploading Large Archives in Parts \(Multipart Upload\)’](#) and [‘Complete Multipart Upload’](#) in the Amazon Glacier Developer Guide .

Parameters

- **checksum** (*string*) – The SHA256 tree hash of the entire archive. It is the tree hash of SHA256 tree hash of the individual parts. If the value you specify in the request does not match the SHA256 tree hash of the final assembled archive as computed by Amazon Glacier, Amazon Glacier returns an error and the request fails.
- **vault_name** (*str*) – The name of the vault.
- **upload_id** (*str*) – The upload ID of the multipart upload.
- **sha256_treehash** (*str*) – The SHA256 tree hash of the entire archive. It is the tree hash of SHA256 tree hash of the individual parts. If the value you specify in the request does not match the SHA256 tree hash of the final assembled archive

as computed by Amazon Glacier, Amazon Glacier returns an error and the request fails.

- **archive_size** (*int*) – The total size, in bytes, of the entire archive. This value should be the sum of all the sizes of the individual parts that you uploaded.

create_vault (*vault_name*)

This operation creates a new vault with the specified name. The name of the vault must be unique within a region for an AWS account. You can create up to 1,000 vaults per account. If you need to create more vaults, contact Amazon Glacier.

You must use the following guidelines when naming a vault.

- Names can be between 1 and 255 characters long.
- Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), and '.' (period).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see ['Access Control Using AWS Identity and Access Management \(IAM\)'](#).

For conceptual information and underlying REST API, go to ['Creating a Vault in Amazon Glacier'](#) and ['Create Vault'](#) in the Amazon Glacier Developer Guide .

Parameters **vault_name** (*string*) – The name of the vault.

delete_archive (*vault_name, archive_id*)

This operation deletes an archive from a vault. Subsequent requests to initiate a retrieval of this archive will fail. Archive retrievals that are in progress for this archive ID may or may not succeed according to the following scenarios:

- If the archive retrieval job is actively preparing the data for download when Amazon Glacier receives the delete archive request, the archival retrieval operation might fail.
- If the archive retrieval job has successfully prepared the archive for download when Amazon Glacier receives the delete archive request, you will be able to download the output.

This operation is idempotent. Attempting to delete an already- deleted archive does not result in an error.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see ['Access Control Using AWS Identity and Access Management \(IAM\)'](#).

For conceptual information and underlying REST API, go to ['Deleting an Archive in Amazon Glacier'](#) and ['Delete Archive'](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **archive_id** (*string*) – The ID of the archive to delete.

delete_vault (*vault_name*)

This operation deletes a vault. Amazon Glacier will delete a vault only if there are no archives in the vault as of the last inventory and there have been no writes to the vault since the last inventory. If either of these conditions is not satisfied, the vault deletion fails (that is, the vault is not removed) and Amazon Glacier returns an error. You can use DescribeVault to return the number of archives in a vault, and you can use ['Initiate a Job \(POST jobs\)'](#) to initiate a new inventory retrieval for a vault. The inventory contains the archive IDs you use to delete archives using ['Delete Archive \(DELETE archive\)'](#).

This operation is idempotent.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit

permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Deleting a Vault in Amazon Glacier’](#) and [‘Delete Vault’](#) in the Amazon Glacier Developer Guide .

Parameters `vault_name` (*string*) – The name of the vault.

delete_vault_notifications (*vault_name*)

This operation deletes the notification configuration set for a vault. The operation is eventually consistent; that is, it might take some time for Amazon Glacier to completely disable the notifications and you might still receive some notifications for a short time after you send the delete request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Configuring Vault Notifications in Amazon Glacier’](#) and [‘Delete Vault Notification Configuration’](#) in the Amazon Glacier Developer Guide.

Parameters `vault_name` (*string*) – The name of the vault.

describe_job (*vault_name, job_id*)

This operation returns information about a job you previously initiated, including the job initiation date, the user who initiated the job, the job status code/message and the Amazon SNS topic to notify after Amazon Glacier completes the job. For more information about initiating a job, see [InitiateJob](#).

This operation enables you to check the status of your job. However, it is strongly recommended that you set up an Amazon SNS topic and specify it in your initiate job request so that Amazon Glacier can notify the topic after it completes the job.

A job ID will not expire for at least 24 hours after Amazon Glacier completes the job.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For information about the underlying REST API, go to [‘Working with Archives in Amazon Glacier’](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **job_id** (*string*) – The ID of the job to describe.

describe_vault (*vault_name*)

This operation returns information about a vault, including the vault’s Amazon Resource Name (ARN), the date the vault was created, the number of archives it contains, and the total size of all the archives in the vault. The number of archives and their total size are as of the last inventory generation. This means that if you add or remove an archive from a vault, and then immediately use [Describe Vault](#), the change in contents will not be immediately reflected. If you want to retrieve the latest inventory of the vault, use [InitiateJob](#). Amazon Glacier generates vault inventories approximately daily. For more information, see [‘Downloading a Vault Inventory in Amazon Glacier’](#).

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Retrieving Vault Metadata in Amazon Glacier’](#) and [‘Describe Vault’](#) in the Amazon Glacier Developer Guide .

Parameters `vault_name` (*string*) – The name of the vault.

get_job_output (*vault_name, job_id, byte_range=None*)

This operation downloads the output of the job you initiated using `InitiateJob`. Depending on the job type you specified when you initiated the job, the output will be either the content of an archive or a vault inventory.

A job ID will not expire for at least 24 hours after Amazon Glacier completes the job. That is, you can download the job output within the 24 hours period after Amazon Glacier completes the job.

If the job output is large, then you can use the *Range* request header to retrieve a portion of the output. This allows you to download the entire output in smaller chunks of bytes. For example, suppose you have 1 GB of job output you want to download and you decide to download 128 MB chunks of data at a time, which is a total of eight Get Job Output requests. You use the following process to download the job output:

1. Download a 128 MB chunk of output by specifying the appropriate byte range using the *Range* header.
2. Along with the data, the response includes a checksum of the payload. You compute the checksum of the payload on the client and compare it with the checksum you received in the response to ensure you received all the expected data.
3. Repeat steps 1 and 2 for all the eight 128 MB chunks of output data, each time specifying the appropriate byte range.
4. After downloading all the parts of the job output, you have a list of eight checksum values. Compute the tree hash of these values to find the checksum of the entire output. Using the Describe Job API, obtain job information of the job that provided you the output. The response includes the checksum of the entire archive stored in Amazon Glacier. You compare this value with the checksum you computed to ensure you have downloaded the entire archive content with no errors.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and the underlying REST API, go to [‘Downloading a Vault Inventory’](#), [‘Downloading an Archive’](#), and [‘Get Job Output’](#).

Parameters

- **account_id** (*string*) – The *AccountId* is the AWS Account ID. You can specify either the AWS Account ID or optionally a ‘-’, in which case Amazon Glacier uses the AWS Account ID associated with the credentials used to sign the request. If you specify your Account ID, do not include hyphens in it.
- **vault_name** (*string*) – The name of the vault.
- **job_id** (*string*) – The job ID whose data is downloaded.
- **byte_range** (*string*) – The range of bytes to retrieve from the output. For example, if you want to download the first 1,048,576 bytes, specify “Range: bytes=0-1048576”. By default, this operation downloads the entire output.

get_vault_notifications (*vault_name*)

This operation retrieves the *notification-configuration* subresource of the specified vault.

For information about setting a notification configuration on a vault, see `SetVaultNotifications`. If a notification configuration for a vault is not set, the operation returns a *404 Not Found* error. For more information about vault notifications, see [‘Configuring Vault Notifications in Amazon Glacier’](#).

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit

permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Configuring Vault Notifications in Amazon Glacier’](#) and [‘Get Vault Notification Configuration’](#) in the Amazon Glacier Developer Guide .

Parameters `vault_name` (*string*) – The name of the vault.

initiate_job (*vault_name, job_data*)

This operation initiates a job of the specified type. In this release, you can initiate a job to retrieve either an archive or a vault inventory (a list of archives in a vault).

Retrieving data from Amazon Glacier is a two-step process:

1. Initiate a retrieval job.
2. After the job completes, download the bytes.

The retrieval request is executed asynchronously. When you initiate a retrieval job, Amazon Glacier creates a job and returns a job ID in the response. When Amazon Glacier completes the job, you can get the job output (archive or inventory data). For information about getting job output, see `GetJobOutput` operation.

The job must complete before you can get its output. To determine when a job is complete, you have the following options:

- **Use Amazon SNS Notification** You can specify an Amazon Simple Notification Service (Amazon SNS) topic to which Amazon Glacier can post a notification after the job is completed. You can specify an SNS topic per job request. The notification is sent only after Amazon Glacier completes the job. In addition to specifying an SNS topic per job request, you can configure vault notifications for a vault so that job notifications are always sent. For more information, see `SetVaultNotifications`.
- **Get job details** You can make a `DescribeJob` request to obtain job status information while a job is in progress. However, it is more efficient to use an Amazon SNS notification to determine when a job is complete.

The information you get via notification is same that you get by calling `DescribeJob`.

If for a specific event, you add both the notification configuration on the vault and also specify an SNS topic in your `initiate_job` request, Amazon Glacier sends both notifications. For more information, see `SetVaultNotifications`.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

About the Vault Inventory

Amazon Glacier prepares an inventory for each vault periodically, every 24 hours. When you initiate a job for a vault inventory, Amazon Glacier returns the last inventory for the vault. The inventory data you get might be up to a day or two days old. Also, the `initiate_inventory_job` might take some time to complete before you can download the vault inventory. So you do not want to retrieve a vault inventory for each vault operation. However, in some scenarios, you might find the vault inventory useful. For example, when you upload an archive, you can provide an archive description but not an archive name. Amazon Glacier provides you a unique archive ID, an opaque string of characters. So, you might maintain your own database that maps archive names to their corresponding Amazon Glacier assigned archive IDs. You might find the vault inventory useful in the event you need to reconcile information in your database with the actual vault inventory.

About Ranged Archive Retrieval

You can initiate an archive retrieval for the whole archive or a range of the archive. In the case of ranged archive retrieval, you specify a byte range to return or the whole archive. The range specified must be megabyte (MB) aligned, that is the range start value must be divisible by 1 MB and range end value plus 1 must be divisible by 1 MB or equal the end of the archive. If the ranged archive retrieval is not megabyte

aligned, this operation returns a 400 response. Furthermore, to ensure you get checksum values for data you download using Get Job Output API, the range must be tree hash aligned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and the underlying REST API, go to [‘Initiate a Job’](#) and [‘Downloading a Vault Inventory’](#).

Parameters

- **account_id** (*string*) – The *AccountId* is the AWS Account ID. You can specify either the AWS Account ID or optionally a ‘-’, in which case Amazon Glacier uses the AWS Account ID associated with the credentials used to sign the request. If you specify your Account ID, do not include hyphens in it.
- **vault_name** (*string*) – The name of the vault.
- **job_parameters** (*dict*) – Provides options for specifying job information. The dictionary can contain the following attributes:
 - **ArchiveId** - The ID of the archive you want to retrieve. This field is required only if the **Type** is set to archive-retrieval.
 - **Description** - The optional description for the job.
 - **Format** - When initiating a job to retrieve a vault inventory, you can optionally add this parameter to specify the output format. Valid values are: CSV|JSON.
 - **SNSTopic** - The Amazon SNS topic ARN where Amazon Glacier sends a notification when the job is completed and the output is ready for you to download.
 - **Type** - The job type. Valid values are: archive-retrieval|inventory-retrieval
 - **RetrievalByteRange** - Optionally specify the range of bytes to retrieve.
 - **InventoryRetrievalParameters: Optional job parameters**
 - * **Format** - The output format, like “JSON”
 - * **StartDate** - ISO8601 starting date string
 - * **EndDate** - ISO8601 ending date string
 - * **Limit** - Maximum number of entries
 - * **Marker** - A unique string used for pagination

initiate_multipart_upload (*vault_name, part_size, description=None*)

This operation initiates a multipart upload. Amazon Glacier creates a multipart upload resource and returns its ID in the response. The multipart upload ID is used in subsequent requests to upload parts of an archive (see `UploadMultipartPart`).

When you initiate a multipart upload, you specify the part size in number of bytes. The part size must be a megabyte (1024 KB) multiplied by a power of 2—for example, 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB.

Every part you upload to this resource (see `UploadMultipartPart`), except the last one, must have the same size. The last one can be the same size or smaller. For example, suppose you want to upload a 16.2 MB

file. If you initiate the multipart upload with a part size of 4 MB, you will upload four parts of 4 MB each and one part of 0.2 MB.

You don't need to know the size of the archive when you start a multipart upload because Amazon Glacier does not require you to specify the overall archive size.

After you complete the multipart upload, Amazon Glacier removes the multipart upload resource referenced by the ID. Amazon Glacier also removes the multipart upload resource if you cancel the multipart upload or it may be removed if there is no activity for a period of 24 hours.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Uploading Large Archives in Parts \(Multipart Upload\)’](#) and [‘Initiate Multipart Upload’](#) in the Amazon Glacier Developer Guide .

The part size must be a megabyte (1024 KB) multiplied by a power of 2, for example, 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB), and so on. The minimum allowable part size is 1 MB, and the maximum is 4 GB (4096 MB).

Parameters

- **vault_name** (*str*) – The name of the vault.
- **description** (*str*) – The archive description that you are uploading in parts.
- **part_size** (*int*) – The size of each part except the last, in bytes. The last part can be smaller than this part size.

list_jobs (*vault_name*, *completed=None*, *status_code=None*, *limit=None*, *marker=None*)

This operation lists jobs for a vault, including jobs that are in-progress and jobs that have recently finished.

Amazon Glacier retains recently completed jobs for a period before deleting them; however, it eventually removes completed jobs. The output of completed jobs can be retrieved. Retaining completed jobs for a period of time after they have completed enables you to get a job output in the event you miss the job completion notification or your first attempt to download it fails. For example, suppose you start an archive retrieval job to download an archive. After the job completes, you start to download the archive but encounter a network error. In this scenario, you can retry and download the archive while the job exists.

To retrieve an archive or retrieve a vault inventory from Amazon Glacier, you first initiate a job, and after the job completes, you download the data. For an archive retrieval, the output is the archive data, and for an inventory retrieval, it is the inventory list. The List Job operation returns a list of these jobs sorted by job initiation time.

This List Jobs operation supports pagination. By default, this operation returns up to 1,000 jobs in the response. You should always check the response for a *marker* at which to continue the list; if there are no more items the *marker* is *null*. To return a list of jobs that begins at a specific job, set the *marker* request parameter to the value you obtained from a previous List Jobs request. You can also limit the number of jobs returned in the response by specifying the *limit* parameter in the request.

Additionally, you can filter the jobs list returned by specifying an optional *statuscode* (InProgress, Succeeded, or Failed) and *completed* (true, false) parameter. The *statuscode* allows you to specify that only jobs that match a specified status are returned. The *completed* parameter allows you to specify that only jobs in a specific completion state are returned.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For the underlying REST API, go to [‘List Jobs’](#) `_`

Parameters

- **vault_name** (*string*) – The name of the vault.
- **limit** (*string*) – Specifies that the response be limited to the specified number of items or fewer. If not specified, the List Jobs operation returns up to 1,000 jobs.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the job at which the listing of jobs should begin. Get the marker value from a previous List Jobs response. You need only include the marker if you are continuing the pagination of results started in a previous List Jobs request.
- **statuscode** (*string*) – Specifies the type of job status to return. You can specify the following values: “InProgress”, “Succeeded”, or “Failed”.
- **completed** (*string*) – Specifies the state of the jobs to return. You can specify *True* or *False*.

list_multipart_uploads (*vault_name, limit=None, marker=None*)

This operation lists in-progress multipart uploads for the specified vault. An in-progress multipart upload is a multipart upload that has been initiated by an `InitiateMultipartUpload` request, but has not yet been completed or aborted. The list returned in the List Multipart Upload response has no guaranteed order.

The List Multipart Uploads operation supports pagination. By default, this operation returns up to 1,000 multipart uploads in the response. You should always check the response for a *marker* at which to continue the list; if there are no more items the *marker* is *null*. To return a list of multipart uploads that begins at a specific upload, set the *marker* request parameter to the value you obtained from a previous List Multipart Upload request. You can also limit the number of uploads returned in the response by specifying the *limit* parameter in the request.

Note the difference between this operation and listing parts (`ListParts`). The List Multipart Uploads operation lists all multipart uploads for a vault and does not require a multipart upload ID. The List Parts operation requires a multipart upload ID since parts are associated with a single upload.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#) `_`.

For conceptual information and the underlying REST API, go to [‘Working with Archives in Amazon Glacier’](#) `_` and [‘List Multipart Uploads’](#) `_` in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **limit** (*string*) – Specifies the maximum number of uploads returned in the response body. If this value is not specified, the List Uploads operation returns up to 1,000 uploads.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the upload at which the listing of uploads should begin. Get the marker value from a previous List Uploads response. You need only include the marker if you are continuing the pagination of results started in a previous List Uploads request.

list_parts (*vault_name, upload_id, limit=None, marker=None*)

This operation lists the parts of an archive that have been uploaded in a specific multipart upload. You can make this request at any time during an in-progress multipart upload before you complete the upload (see `CompleteMultipartUpload`). List Parts returns an error for completed uploads. The list returned in the List Parts response is sorted by part range.

The List Parts operation supports pagination. By default, this operation returns up to 1,000 uploaded parts in the response. You should always check the response for a *marker* at which to continue the list; if there are no more items the *marker* is *null*. To return a list of parts that begins at a specific part, set the *marker* request parameter to the value you obtained from a previous List Parts request. You can also limit the number of parts returned in the response by specifying the *limit* parameter in the request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and the underlying REST API, go to [‘Working with Archives in Amazon Glacier’](#) and [‘List Parts’](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **upload_id** (*string*) – The upload ID of the multipart upload.
- **marker** (*string*) – An opaque string used for pagination. This value specifies the part at which the listing of parts should begin. Get the marker value from the response of a previous List Parts response. You need only include the marker if you are continuing the pagination of results started in a previous List Parts request.
- **limit** (*string*) – Specifies the maximum number of parts returned in the response body. If this value is not specified, the List Parts operation returns up to 1,000 uploads.

list_vaults (*limit=None, marker=None*)

This operation lists all vaults owned by the calling user's account. The list returned in the response is ASCII-sorted by vault name.

By default, this operation returns up to 1,000 items. If there are more vaults to list, the response *marker* field contains the vault Amazon Resource Name (ARN) at which to continue the list with a new List Vaults request; otherwise, the *marker* field is *null*. To return a list of vaults that begins at a specific vault, set the *marker* request parameter to the vault ARN you obtained from a previous List Vaults request. You can also limit the number of vaults returned in the response by specifying the *limit* parameter in the request.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Retrieving Vault Metadata in Amazon Glacier’](#) and [‘List Vaults’](#) in the Amazon Glacier Developer Guide .

Parameters

- **marker** (*string*) – A string used for pagination. The marker specifies the vault ARN after which the listing of vaults should begin.
- **limit** (*string*) – The maximum number of items returned in the response. If you don't specify a value, the List Vaults operation returns up to 1,000 items.

make_request (*verb, resource, headers=None, data=”, ok_responses=(200,), params=None, sender=None, response_headers=None*)

Makes a request to the server, with stock multiple-retry logic.

set_vault_notifications (*vault_name, notification_config*)

This operation configures notifications that will be sent when specific events happen to a vault. By default, you don't get any notifications.

To configure vault notifications, send a PUT request to the *notification-configuration* subresource of the vault. The request should include a JSON document that provides an Amazon SNS topic and specific events for which you want Amazon Glacier to send notifications to the topic.

Amazon SNS topics must grant permission to the vault to be allowed to publish notifications to the topic. You can configure a vault to publish a notification for the following vault events:

- **ArchiveRetrievalCompleted** This event occurs when a job that was initiated for an archive retrieval is completed (InitiateJob). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from DescribeJob.
- **InventoryRetrievalCompleted** This event occurs when a job that was initiated for an inventory retrieval is completed (InitiateJob). The status of the completed job can be “Succeeded” or “Failed”. The notification sent to the SNS topic is the same output as returned from DescribeJob.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Configuring Vault Notifications in Amazon Glacier’](#) and [‘Set Vault Notification Configuration’](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*string*) – The name of the vault.
- **vault_notification_config** (*dict*) – Provides options for specifying notification configuration.

The format of the dictionary is:

```
{‘SNSTopic’: ‘mytopic’, ‘Events’: [event1,...]}
```

upload_archive (*vault_name, archive, linear_hash, tree_hash, description=None*)

This operation adds an archive to a vault. This is a synchronous operation, and for a successful upload, your data is durably persisted. Amazon Glacier returns the archive ID in the *x-amz-archive-id* header of the response.

You must use the archive ID to access your data in Amazon Glacier. After you upload an archive, you should save the archive ID returned so that you can retrieve or delete the archive later. Besides saving the archive ID, you can also index it and give it a friendly name to allow for better searching. You can also use the optional archive description field to specify how the archive is referred to in an external index of archives, such as you might create in Amazon DynamoDB. You can also get the vault inventory to obtain a list of archive IDs in a vault. For more information, see InitiateJob.

You must provide a SHA256 tree hash of the data you are uploading. For information about computing a SHA256 tree hash, see [‘Computing Checksums’](#).

You can optionally specify an archive description of up to 1,024 printable ASCII characters. You can get the archive description when you either retrieve the archive or get the vault inventory. For more information, see InitiateJob. Amazon Glacier does not interpret the description in any way. An archive description does not need to be unique. You cannot use the description to retrieve or sort the archive list.

Archives are immutable. After you upload an archive, you cannot edit the archive or its description.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don’t have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see [‘Access Control Using AWS Identity and Access Management \(IAM\)’](#).

For conceptual information and underlying REST API, go to [‘Uploading an Archive in Amazon Glacier’](#) and [‘Upload Archive’](#) in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*str*) – The name of the vault
- **archive** (*bytes*) – The data to upload.
- **linear_hash** (*str*) – The SHA256 checksum (a linear hash) of the payload.
- **tree_hash** (*str*) – The user-computed SHA256 tree hash of the payload. For more information on computing the tree hash, see <http://goo.gl/u7chF>.
- **description** (*str*) – The optional description of the archive you are uploading.

upload_part (*vault_name, upload_id, linear_hash, tree_hash, byte_range, part_data*)

This operation uploads a part of an archive. You can upload archive parts in any order. You can also upload them in parallel. You can upload up to 10,000 parts for a multipart upload.

Amazon Glacier rejects your upload part request if any of the following conditions is true:

- ****SHA256 tree hash does not match****To ensure that part data is not corrupted in transmission, you compute a SHA256 tree hash of the part and include it in your request. Upon receiving the part data, Amazon Glacier also computes a SHA256 tree hash. If these hash values don't match, the operation fails. For information about computing a SHA256 tree hash, see '[Computing Checksums](#)'.
- ****Part size does not match****The size of each part except the last must match the size specified in the corresponding InitiateMultipartUpload request. The size of the last part must be the same size as, or smaller than, the specified size. If you upload a part whose size is smaller than the part size you specified in your initiate multipart upload request and that part is not the last part, then the upload part request will succeed. However, the subsequent Complete Multipart Upload request will fail.
- ****Range does not align****The byte range value in the request does not align with the part size specified in the corresponding initiate request. For example, if you specify a part size of 4194304 bytes (4 MB), then 0 to 4194303 bytes (4 MB - 1) and 4194304 (4 MB) to 8388607 (8 MB - 1) are valid part ranges. However, if you set a range value of 2 MB to 6 MB, the range does not align with the part size and the upload will fail.

This operation is idempotent. If you upload the same part multiple times, the data included in the most recent request overwrites the previously uploaded data.

An AWS account has full permission to perform all operations (actions). However, AWS Identity and Access Management (IAM) users don't have any permissions by default. You must grant them explicit permission to perform specific actions. For more information, see '[Access Control Using AWS Identity and Access Management \(IAM\)](#)'.

For conceptual information and underlying REST API, go to '[Uploading Large Archives in Parts \(Multipart Upload\)](#)' and 'Upload Part' in the Amazon Glacier Developer Guide .

Parameters

- **vault_name** (*str*) – The name of the vault.
- **linear_hash** (*str*) – The SHA256 checksum (a linear hash) of the payload.
- **tree_hash** (*str*) – The user-computed SHA256 tree hash of the payload. For more information on computing the tree hash, see <http://goo.gl/u7chF>.
- **upload_id** (*str*) – The unique ID associated with this upload operation.
- **byte_range** (*tuple of ints*) – Identifies the range of bytes in the assembled archive that will be uploaded in this part. Amazon Glacier uses this information to assemble the archive in the proper sequence. The format of this header follows RFC 2616. An example header is Content-Range:bytes 0-4194303/*.
- **part_data** (*bytes*) – The data to be uploaded for the part

4.135.3 boto.glacier.layer2

class `boto.glacier.layer2.Layer2` (**args, **kwargs*)
Provides a more pythonic and friendly interface to Glacier based on Layer1

create_vault (*name*)

Creates a vault.

Parameters `name` (*str*) – The name of the vault

Return type `boto.glacier.vault.Vault`

Returns A Vault object representing the vault.

delete_vault (*name*)

Delete a vault.

This operation deletes a vault. Amazon Glacier will delete a vault only if there are no archives in the vault as per the last inventory and there have been no writes to the vault since the last inventory. If either of these conditions is not satisfied, the vault deletion fails (that is, the vault is not removed) and Amazon Glacier returns an error.

This operation is idempotent, you can send the same request multiple times and it has no further effect after the first time Amazon Glacier delete the specified vault.

Parameters `vault_name` (*str*) – The name of the vault to delete.

get_vault (*name*)

Get an object representing a named vault from Glacier. This operation does not check if the vault actually exists.

Parameters `name` (*str*) – The name of the vault

Return type `boto.glacier.vault.Vault`

Returns A Vault object representing the vault.

list_vaults ()

Return a list of all vaults associated with the account ID.

Return type List of `boto.glacier.vault.Vault`

Returns A list of Vault objects.

4.135.4 boto.glacier.vault

class `boto.glacier.vault.Vault` (*layer1, response_data=None*)

DefaultPartSize = 4194304

ResponseDataElements = (('VaultName', 'name', None), ('VaultARN', 'arn', None), ('Creat

SingleOperationThreshold = 104857600

concurrent_create_archive_from_file (*filename, description, **kwargs*)

Create a new archive from a file and upload the given file.

This is a convenience method around the `boto.glacier.concurrent.ConcurrentUploader` class. This method will perform a multipart upload and upload the parts of the file concurrently.

Parameters

- **filename** (*str*) – A filename to upload

- **kwargs** – Additional kwargs to pass through to `boto.glacier.concurrent.ConcurrentUploader`. You can pass any argument besides the `api` and `vault_name` param (these arguments are already passed to the `ConcurrentUploader` for you).

Raises `boto.glacier.exception.UploadArchiveError` is an error occurs during the upload process.

Return type `str`

Returns The archive id of the newly created archive

create_archive_from_file (`filename=None`, `file_obj=None`, `description=None`, `upload_id_callback=None`)

Create a new archive and upload the data from the given file or file-like object.

Parameters

- **filename** (`str`) – A filename to upload
- **file_obj** (`file`) – A file-like object to upload
- **description** (`str`) – An optional description for the archive.
- **upload_id_callback** (`function`) – if set, call with the `upload_id` as the only parameter when it becomes known, to enable future calls to `resume_archive_from_file` in case resume is needed.

Return type `str`

Returns The archive id of the newly created archive

create_archive_writer (`part_size=4194304`, `description=None`)

Create a new archive and begin a multi-part upload to it. Returns a file-like object to which the data for the archive can be written. Once all the data is written the file-like object should be closed, you can then call the `get_archive_id` method on it to get the ID of the created archive.

Parameters

- **part_size** (`int`) – The part size for the multipart upload.
- **description** (`str`) – An optional description for the archive.

Return type `boto.glacier.writer.Writer`

Returns A `Writer` object that to which the archive data should be written.

delete ()

Delete's this vault. WARNING!

delete_archive (`archive_id`)

This operation deletes an archive from the vault.

Parameters **archive_id** (`str`) – The ID for the archive to be deleted.

get_job (`job_id`)

Get an object representing a job in progress.

Parameters **job_id** (`str`) – The ID of the job

Return type `boto.glacier.job.Job`

Returns A `Job` object representing the job.

list_all_parts (`upload_id`)

Automatically make and combine multiple calls to `list_parts`.

Call `list_parts` as necessary, combining the results in case multiple calls were required to get data on all available parts.

list_jobs (*completed=None, status_code=None*)

Return a list of Job objects related to this vault.

Parameters

- **completed** (*boolean*) – Specifies the state of the jobs to return. If a value of True is passed, only completed jobs will be returned. If a value of False is passed, only uncompleted jobs will be returned. If no value is passed, all jobs will be returned.
- **status_code** (*string*) – Specifies the type of job status to return. Valid values are: InProgress|Succeeded|Failed. If not specified, jobs with all status codes are returned.

Return type list of *boto.glacier.job.Job*

Returns A list of Job objects related to this vault.

resume_archive_from_file (*upload_id, filename=None, file_obj=None*)

Resume upload of a file already part-uploaded to Glacier.

The resumption of an upload where the part-uploaded section is empty is a valid degenerate case that this function can handle.

One and only one of filename or file_obj must be specified.

Parameters

- **upload_id** (*str*) – existing Glacier upload id of upload being resumed.
- **filename** (*str*) – file to open for resume
- **fobj** (*file*) – file-like object containing local data to resume. This must read from the start of the entire upload, not just from the point being resumed. Use fobj.seek(0) to achieve this if necessary.

Return type *str*

Returns The archive id of the newly created archive

retrieve_archive (*archive_id, sns_topic=None, description=None*)

Initiate a archive retrieval job to download the data from an archive. You will need to wait for the notification from Amazon (via SNS) before you can actually download the data, this takes around 4 hours.

Parameters

- **archive_id** (*str*) – The id of the archive
- **description** (*str*) – An optional description for the job.
- **sns_topic** (*str*) – The Amazon SNS topic ARN where Amazon Glacier sends notification when the job is completed and the output is ready for you to download.

Return type *boto.glacier.job.Job*

Returns A Job object representing the retrieval job.

retrieve_inventory (*sns_topic=None, description=None, byte_range=None, start_date=None, end_date=None, limit=None*)

Initiate a inventory retrieval job to list the items in the vault. You will need to wait for the notification from Amazon (via SNS) before you can actually download the data, this takes around 4 hours.

Parameters

- **description** (*str*) – An optional description for the job.

- **sns_topic** (*str*) – The Amazon SNS topic ARN where Amazon Glacier sends notification when the job is completed and the output is ready for you to download.
- **byte_range** (*str*) – Range of bytes to retrieve.
- **start_date** (*DateTime*) – Beginning of the date range to query.
- **end_date** (*DateTime*) – End of the date range to query.
- **limit** (*int*) – Limits the number of results returned.

Return type *str*

Returns The ID of the job

retrieve_inventory_job (***kwargs*)

Identical to `retrieve_inventory`, but returns a `Job` instance instead of just the job ID.

Parameters

- **description** (*str*) – An optional description for the job.
- **sns_topic** (*str*) – The Amazon SNS topic ARN where Amazon Glacier sends notification when the job is completed and the output is ready for you to download.
- **byte_range** (*str*) – Range of bytes to retrieve.
- **start_date** (*DateTime*) – Beginning of the date range to query.
- **end_date** (*DateTime*) – End of the date range to query.
- **limit** (*int*) – Limits the number of results returned.

Return type *boto.glacier.job.Job*

Returns A `Job` object representing the retrieval job.

upload_archive (*filename, description=None*)

Adds an archive to a vault. For archives greater than 100MB the multipart upload will be used.

Parameters

- **file** (*str*) – A filename to upload
- **description** (*str*) – An optional description for the archive.

Return type *str*

Returns The archive id of the newly created archive

4.135.5 boto.glacier.job

```
class boto.glacier.job.Job (vault, response_data=None)
```

```
    DefaultPartSize = 4194304
```

```
    ResponseDataElements = (('Action', 'action', None), ('ArchiveId', 'archive_id', None),
```

```
    download_to_file (filename, chunk_size=4194304, verify_hashes=True, retry_exceptions=(<class
        'socket.error'>,)

```

```
    Download an archive to a file by name.
```

Parameters

- **filename** (*str*) – The name of the file where the archive contents will be saved.
- **chunk_size** (*int*) – The chunk size to use when downloading the archive.
- **verify_hashes** (*bool*) – Indicates whether or not to verify the tree hashes for each downloaded chunk.

```
download_to_fileobj (output_file, chunk_size=4194304, verify_hashes=True,  
                    retry_exceptions=(<class 'socket.error'>,))
```

Download an archive to a file object.

Parameters

- **output_file** (*file*) – The file object where the archive contents will be saved.
- **chunk_size** (*int*) – The chunk size to use when downloading the archive.
- **verify_hashes** (*bool*) – Indicates whether or not to verify the tree hashes for each downloaded chunk.

```
get_output (byte_range=None, validate_checksum=False)
```

This operation downloads the output of the job. Depending on the job type you specified when you initiated the job, the output will be either the content of an archive or a vault inventory.

You can download all the job output or download a portion of the output by specifying a byte range. In the case of an archive retrieval job, depending on the byte range you specify, Amazon Glacier returns the checksum for the portion of the data. You can compute the checksum on the client and verify that the values match to ensure the portion you downloaded is the correct data.

Parameters

- **range** – A tuple of integer specifying the slice (in bytes) of the archive you want to receive
- **validate_checksum** (*bool*) – Specify whether or not to validate the associate tree hash. If the response does not contain a TreeHash, then no checksum will be verified.

4.135.6 boto.glacier.writer

```
class boto.glacier.writer.Writer (vault, upload_id, part_size, chunk_size=1048576)
```

Presents a file-like object for writing to a Amazon Glacier Archive. The data is written using the multi-part upload API.

```
close ()
```

```
current_tree_hash
```

Returns the current tree hash for the data that's been written **so far**.

Only once the writing is complete is the final tree hash returned.

```
current_uploaded_size
```

Returns the current uploaded size for the data that's been written **so far**.

Only once the writing is complete is the final uploaded size returned.

```
get_archive_id ()
```

```
upload_id
```

```
vault
```

```
write (data)
```

`boto.glacier.writer.generate_parts_from_fobj` (*fobj*, *part_size*)

`boto.glacier.writer.resume_file_upload` (*vault*, *upload_id*, *part_size*, *fobj*, *part_hash_map*,
chunk_size=1048576)

Resume upload of a file already part-uploaded to Glacier.

The resumption of an upload where the part-uploaded section is empty is a valid degenerate case that this function can handle. In this case, *part_hash_map* should be an empty dict.

Parameters

- **vault** – `boto.glacier.vault.Vault` object.
- **upload_id** – existing Glacier upload id of upload being resumed.
- **part_size** – part size of existing upload.
- **fobj** – file object containing local data to resume. This must read from the start of the entire upload, not just from the point being resumed. Use `fobj.seek(0)` to achieve this if necessary.
- **part_hash_map** – {*part_index*: *part_tree_hash*, ... } of data already uploaded. Each supplied *part_tree_hash* will be verified and the part re-uploaded if there is a mismatch.
- **chunk_size** – chunk size of tree hash calculation. This must be 1 MiB for Amazon.

4.135.7 boto.glacier.concurrent

class `boto.glacier.concurrent.ConcurrentDownloader` (*job*, *part_size=4194304*,
num_threads=10)

Concurrently download an archive from glacier.

This class uses a thread pool to concurrently download an archive from glacier.

The threadpool is completely managed by this class and is transparent to the users of this class.

Parameters

- **job** – A layer2 job object for archive retrieval object.
- **part_size** – The size, in bytes, of the chunks to use when uploading the archive parts. The part size must be a megabyte multiplied by a power of two.

download (*filename*)

Concurrently download an archive.

Parameters **filename** (*str*) – The filename to download the archive to

class `boto.glacier.concurrent.ConcurrentTransferer` (*part_size=4194304*,
num_threads=10)

class `boto.glacier.concurrent.ConcurrentUploader` (*api*, *vault_name*, *part_size=4194304*,
num_threads=10)

Concurrently upload an archive to glacier.

This class uses a thread pool to concurrently upload an archive to glacier using the multipart upload API.

The threadpool is completely managed by this class and is transparent to the users of this class.

Parameters

- **api** (`boto.glacier.layer1.Layer1`) – A layer1 glacier object.
- **vault_name** (*str*) – The name of the vault.
- **part_size** (*int*) – The size, in bytes, of the chunks to use when uploading the archive parts. The part size must be a megabyte multiplied by a power of two.

- **num_threads** (*int*) – The number of threads to spawn for the thread pool. The number of threads will control how much parts are being concurrently uploaded.

upload (*filename*, *description=None*)

Concurrently create an archive.

The *part_size* value specified when the class was constructed will be used *unless* it is smaller than the minimum required part size needed for the size of the given file. In that case, the part size used will be the minimum part size required to properly upload the given file.

Parameters

- **file** (*str*) – The filename to upload
- **description** (*str*) – The description of the archive.

Return type *str*

Returns The archive id of the newly created archive.

```
class boto.glacier.concurrent.DownloadWorkerThread (job, worker_queue, re-
result_queue, num_retries=5,
time_between_retries=5,
retry_exceptions=<type 'ex-
ceptions.Exception'>)
```

Individual download thread that will download parts of the file from Glacier. Parts to download stored in work queue.

Parts download to a temp dir with each part a separate file

Parameters

- **job** – Glacier job object
- **work_queue** – A queue of tuples which include the *part_number* and *part_size*
- **result_queue** – A priority queue of tuples which include the *part_number* and the path to the temp file that holds that part's data.

```
class boto.glacier.concurrent.TransferThread (worker_queue, result_queue)
```

run ()

Method representing the thread's activity.

You may override this method in a subclass. The standard *run()* method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the *args* and *kwargs* arguments, respectively.

```
class boto.glacier.concurrent.UploadWorkerThread (api, vault_name, filename, up-
load_id, worker_queue, re-
result_queue, num_retries=5,
time_between_retries=5,
retry_exceptions=<type 'excep-
tions.Exception'>)
```

4.135.8 boto.glacier.exceptions

```
exception boto.glacier.exceptions.ArchiveError
```

```
exception boto.glacier.exceptions.DownloadArchiveError
```

```
exception boto.glacier.exceptions.TreeHashDoesNotMatchError
```

exception boto.glacier.exceptions.**UnexpectedHTTPResponseError** (*expected_responses,*
response)

exception boto.glacier.exceptions.**UploadArchiveError**

4.136 GS

4.136.1 boto.gs.acl

class boto.gs.acl.**ACL** (*parent=None*)

acl

add_email_grant (*permission, email_address*)

add_group_email_grant (*permission, email_address*)

add_group_grant (*permission, group_id*)

add_user_grant (*permission, user_id*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

boto.gs.acl.**CannedACLStrings** = ['private', 'public-read', 'project-private', 'public-read-
A list of Google Cloud Storage predefined (canned) ACL strings.

class boto.gs.acl.**Entries** (*parent=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class boto.gs.acl.**Entry** (*scope=None, type=None, id=None, name=None, email_address=None,*
domain=None, permission=None)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class boto.gs.acl.**Scope** (*parent, type=None, id=None, name=None, email_address=None, do-*
main=None)

ALLOWED_SCOPE_TYPE_SUB_ELEMS = {'allauthenticatedusers': [], 'allusers': [], 'groupby

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

boto.gs.acl.**SupportedPermissions** = ['READ', 'WRITE', 'FULL_CONTROL']
A list of supported ACL permissions.

4.136.2 boto.gs.bucket

```
class boto.gs.bucket.Bucket (connection=None, name=None, key_class=<class
                             'boto.gs.key.Key'>)
```

Represents a Google Cloud Storage bucket.

```
BillingBody = '<?xml version="1.0" encoding="UTF-8"?>\n<BillingConfiguration><Requester
```

```
EncryptionConfigBody = '<?xml version="1.0" encoding="UTF-8"?>\n<EncryptionConfigurati
```

```
EncryptionConfigDefaultKeyNameFragment = '<DefaultKmsKeyName>%s</DefaultKmsKeyName>'
```

```
StorageClassBody = '<?xml version="1.0" encoding="UTF-8"?>\n<StorageClass>%s</StorageC
```

```
add_email_grant (permission, email_address, recursive=False, headers=None)
```

Convenience method that provides a quick way to add an email grant to a bucket. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GCS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ, WRITE, FULL_CONTROL).
- **email_address** (*string*) – The email address associated with the GS account your are granting the permission to.
- **recursive** (*bool*) – A boolean value to controls whether the call will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!

```
add_group_email_grant (permission, email_address, recursive=False, headers=None)
```

Convenience method that provides a quick way to add an email group grant to a bucket. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GCS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: READ|WRITE|FULL_CONTROL See <http://code.google.com/apis/storage/docs/developer-guide.html#authorization> for more details on permissions.
- **email_address** (*string*) – The email address associated with the Google Group to which you are granting the permission.
- **recursive** (*bool*) – A boolean value to controls whether the call will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!

```
add_user_grant (permission, user_id, recursive=False, headers=None)
```

Convenience method that provides a quick way to add a canonical user grant to a bucket. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUTs the new ACL back to GCS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ|WRITE|FULL_CONTROL)
- **user_id** (*string*) – The canonical user id associated with the GS account you are granting the permission to.

- **recursive** (*bool*) – A boolean value to controls whether the call will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!

cancel_multipart_upload (*key_name, upload_id, headers=None*)

To verify that all parts have been removed, so you don't get charged for the part storage, you should call the List Parts operation and ensure the parts list is empty.

complete_multipart_upload (*key_name, upload_id, xml_body, headers=None*)

Complete a multipart upload operation.

configure_billing (*requester_pays=False, headers=None*)

Configure billing for this bucket.

Parameters

- **requester_pays** (*bool*) – If set to True, enables requester pays on this bucket. If set to False, disables requester pays.
- **headers** (*dict*) – Additional headers to send with the request.

configure_lifecycle (*lifecycle_config, headers=None*)

Configure lifecycle for this bucket.

Parameters lifecycle_config (*boto.gs.lifecycle.LifecycleConfig*) –

The lifecycle configuration you want to configure for this bucket.

configure_versioning (*enabled, headers=None*)

Configure versioning for this bucket.

Parameters

- **enabled** (*bool*) – If set to True, enables versioning on this bucket. If set to False, disables versioning.
- **headers** (*dict*) – Additional headers to send with the request.

configure_website (*main_page_suffix=None, error_key=None, headers=None*)

Configure this bucket to act as a website

Parameters

- **main_page_suffix** (*str*) – Suffix that is appended to a request that is for a “directory” on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html). The suffix must not be empty and must not include a slash character. This parameter is optional and the property is disabled if excluded.
- **error_key** (*str*) – The object key name to use when a 400 error occurs. This parameter is optional and the property is disabled if excluded.
- **headers** (*dict*) – Additional headers to send with the request.

copy_key (*new_key_name, src_bucket_name, src_key_name, metadata=None, src_version_id=None, storage_class='STANDARD', preserve_acl=False, encrypt_key=False, headers=None, query_args=None, src_generation=None*)

Create a new key in the bucket by copying an existing key.

Parameters

- **new_key_name** (*string*) – The name of the new key
- **src_bucket_name** (*string*) – The name of the source bucket
- **src_key_name** (*string*) – The name of the source key

- **src_generation** (*int*) – The generation number of the source key to copy. If not specified, the latest generation is copied.
- **metadata** (*dict*) – Metadata to be associated with new key. If metadata is supplied, it will replace the metadata of the source key being copied. If no metadata is supplied, the source key’s metadata will be copied to the new key.
- **version_id** (*string*) – Unused in this subclass.
- **storage_class** (*string*) – The storage class of the new key. By default, the new key will use the standard storage class. Possible values are: STANDARD | DURABLE_REDUCED_AVAILABILITY
- **preserve_acl** (*bool*) – If True, the ACL from the source key will be copied to the destination key. If False, the destination key will have the default ACL. Note that preserving the ACL in the new key object will require two additional API calls to GCS, one to retrieve the current ACL and one to set that ACL on the new object. If you don’t care about the ACL (or if you have a default ACL set on the bucket), a value of False will be significantly more efficient.
- **encrypt_key** (*bool*) – Included for compatibility with S3. This argument is ignored.
- **headers** (*dict*) – A dictionary of header name/value pairs.
- **query_args** (*string*) – A string of additional querystring arguments to append to the request

Return type *boto.gs.key.Key*

Returns An instance of the newly created key object

delete (*headers=None*)

delete_cors (*headers=None*)

Removes all CORS configuration from the bucket.

delete_key (*key_name, headers=None, version_id=None, mfa_token=None, generation=None*)

Deletes a key from the bucket.

Parameters

- **key_name** (*string*) – The key name to delete
- **headers** (*dict*) – A dictionary of header name/value pairs.
- **version_id** (*string*) – Unused in this subclass.
- **mfa_token** (*tuple or list of strings*) – Unused in this subclass.
- **generation** (*int*) – The generation number of the key to delete. If not specified, the latest generation number will be deleted.

Return type *boto.gs.key.Key*

Returns A key object holding information on what was deleted.

delete_keys (*keys, quiet=False, mfa_token=None, headers=None*)

Deletes a set of keys using S3’s Multi-object delete API. If a VersionID is specified for that key then that version is removed. Returns a MultiDeleteResult Object, which contains Deleted and Error elements for each key you ask to delete.

Parameters

- **keys** (*list*) – A list of either key_names or (key_name, versionid) pairs or a list of Key instances.

- **quiet** (*boolean*) – In quiet mode the response includes only keys where the delete operation encountered an error. For a successful deletion, the operation does not return any information about the delete in the response body.
- **mfa_token** (*tuple or list of strings*) – A tuple or list consisting of the serial number from the MFA device and the current value of the six-digit token associated with the device. This value is required anytime you are deleting versioned objects from a bucket that has the MFADelete option on the bucket.

Returns An instance of MultiDeleteResult

delete_lifecycle_configuration (*headers=None*)

Removes all lifecycle configuration from the bucket.

delete_policy (*headers=None*)

delete_tags (*headers=None*)

delete_website_configuration (*headers=None*)

Remove the website configuration from this bucket.

Parameters headers (*dict*) – Additional headers to send with the request.

disable_logging (*headers=None*)

Disable logging on this bucket.

Parameters headers (*dict*) – Additional headers to send with the request.

enable_logging (*target_bucket, target_prefix=None, headers=None*)

Enable logging on a bucket.

Parameters

- **target_bucket** (*bucket or string*) – The bucket to log to.
- **target_prefix** (*string*) – The prefix which should be prepended to the generated log files written to the target_bucket.
- **headers** (*dict*) – Additional headers to send with the request.

generate_url (*expires_in, method='GET', headers=None, force_http=False, response_headers=None, expires_in_absolute=False*)

get_acl (*key_name="", headers=None, version_id=None, generation=None*)

Returns the ACL of the bucket or an object in the bucket.

Parameters

- **key_name** (*str*) – The name of the object to get the ACL for. If not specified, the ACL for the bucket will be returned.
- **headers** (*dict*) – Additional headers to set during the request.
- **version_id** (*string*) – Unused in this subclass.
- **generation** (*int*) – If specified, gets the ACL for a specific generation of a versioned object. If not specified, the current version is returned. This parameter is only valid when retrieving the ACL of an object, not a bucket.

Return type *gs.acl.ACL*

get_all_keys (*headers=None, **params*)

A lower-level method for listing contents of a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the list method.

Parameters

- **max_keys** (*int*) – The maximum number of keys to retrieve

- **prefix** (*string*) – The prefix of the keys you want to retrieve
- **marker** (*string*) – The “marker” of where you are in the result set
- **delimiter** (*string*) – If this optional, Unicode string parameter is included with your request, then keys that contain the same string between the prefix and the first occurrence of the delimiter will be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

Return type *ResultSet*

Returns The result from S3 listing the keys requested

get_all_multipart_uploads (*headers=None, **params*)

A lower-level, version-aware method for listing active MultiPart uploads for a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the list method.

Parameters

- **max_uploads** (*int*) – The maximum number of uploads to retrieve. Default value is 1000.
 - **key_marker** (*string*) – Together with `upload_id_marker`, this parameter specifies the multipart upload after which listing should begin. If `upload_id_marker` is not specified, only the keys lexicographically greater than the specified `key_marker` will be included in the list.
- If `upload_id_marker` is specified, any multipart uploads for a key equal to the `key_marker` might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified `upload_id_marker`.
- **upload_id_marker** (*string*) – Together with `key-marker`, specifies the multipart upload after which listing should begin. If `key_marker` is not specified, the `upload_id_marker` parameter is ignored. Otherwise, any multipart uploads for a key equal to the `key_marker` might be included in the list only if they have an upload ID lexicographically greater than the specified `upload_id_marker`.
 - **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

- **delimiter** (*string*) – Character you use to group keys. All keys that contain the same string between the prefix, if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, CommonPrefixes. If you don't specify the prefix parameter, then the substring starts at the

beginning of the key. The keys that are grouped under CommonPrefixes result element are not returned elsewhere in the response.

- **prefix** (*string*) – Lists in-progress uploads only for those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different grouping of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)

Return type *ResultSet*

Returns The result from S3 listing the uploads requested

get_all_versions (*headers=None, **params*)

A lower-level, version-aware method for listing contents of a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the list method.

Parameters

- **max_keys** (*int*) – The maximum number of keys to retrieve
- **prefix** (*string*) – The prefix of the keys you want to retrieve
- **key_marker** (*string*) – The “marker” of where you are in the result set with respect to keys.
- **version_id_marker** (*string*) – The “marker” of where you are in the result set with respect to version-id's.
- **delimiter** (*string*) – If this optional, Unicode string parameter is included with your request, then keys that contain the same string between the prefix and the first occurrence of the delimiter will be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: url

Return type *ResultSet*

Returns The result from S3 listing the keys requested

get_billing_config (*headers=None*)

Returns the current status of billing configuration on the bucket.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type *dict*

Returns

A dictionary containing the parsed XML response from GCS. The overall structure is:

- BillingConfiguration
 - RequesterPays: Enabled/Disabled.

get_billing_configuration_with_xml (*headers=None*)

Returns the current status of billing configuration on the bucket as unparsed XML.

Parameters `headers` (*dict*) – Additional headers to send with the request.

Return type 2-Tuple

Returns

2-tuple containing:

1. A dictionary containing the parsed XML response from GCS. The overall structure is:
 - `BillingConfiguration`
 - `RequesterPays`: Enabled/Disabled.
2. Unparsed XML describing the bucket's website configuration.

get_cors (*headers=None*)

Returns a bucket's CORS XML document.

Parameters `headers` (*dict*) – Additional headers to send with the request.

Return type `Cors`

get_cors_xml (*headers=None*)

Returns the current CORS configuration on the bucket as an XML document.

get_def_acl (*headers=None*)

Returns the bucket's default ACL.

Parameters `headers` (*dict*) – Additional headers to set during the request.

Return type `gs.acl.ACL`

get_encryption_config (*headers=None*)

Returns a bucket's EncryptionConfig.

Parameters `headers` (*dict*) – Additional headers to send with the request.

Return type `EncryptionConfig`

get_key (*key_name, headers=None, version_id=None, response_headers=None, generation=None*)

Returns a Key instance for an object in this bucket.

Note that this method uses a HEAD request to check for the existence of the key.

Parameters

- **key_name** (*string*) – The name of the key to retrieve
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/06N3b> for details.
- **version_id** (*string*) – Unused in this subclass.
- **generation** (*int*) – A specific generation number to fetch the key at. If not specified, the latest generation is fetched.

Return type `boto.gs.key.Key`

Returns A Key object from this bucket.

get_lifecycle_config (*headers=None*)

Returns the current lifecycle configuration on the bucket.

Return type `boto.gs.lifecycle.LifecycleConfig`

Returns A LifecycleConfig object that describes all current lifecycle rules in effect for the bucket.

get_location (*headers=None*)

Returns the LocationConstraint for the bucket.

Return type `str`

Returns The LocationConstraint for the bucket or the empty string if no constraint was specified when bucket was created.

get_logging_config (*headers=None*)

Returns the current status of logging configuration on the bucket.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type `dict`

Returns

A dictionary containing the parsed XML response from GCS. The overall structure is:

- Logging
 - LogObjectPrefix: Prefix that is prepended to log objects.
 - LogBucket: Target bucket for log objects.

get_logging_config_with_xml (*headers=None*)

Returns the current status of logging configuration on the bucket as unparsed XML.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type 2-Tuple

Returns

2-tuple containing:

1. A dictionary containing the parsed XML response from GCS. The overall structure is:
 - Logging
 - LogObjectPrefix: Prefix that is prepended to log objects.
 - LogBucket: Target bucket for log objects.
2. Unparsed XML describing the bucket's logging configuration.

get_logging_status (*headers=None*)

Get the logging status for this bucket.

Return type `boto.s3.bucketlogging.BucketLogging`

Returns A BucketLogging object for this bucket.

get_policy (*headers=None*)

Returns the JSON policy associated with the bucket. The policy is returned as an uninterpreted JSON string.

get_request_payment (*headers=None*)

get_storage_class (*headers=None*)

Returns the StorageClass for the bucket.

Return type `str`

Returns The StorageClass for the bucket.

get_subresource (*subresource*, *key_name=""*, *headers=None*, *version_id=None*)

Get a subresource for a bucket or key.

Parameters

- **subresource** (*string*) – The subresource to get.
- **key_name** (*string*) – The key to operate on, or None to operate on the bucket.
- **headers** (*dict*) – Additional HTTP headers to include in the request.
- **src_version_id** (*string*) – Optional. The version id of the key to operate on. If not specified, operate on the newest version.

Return type *string*

Returns The value of the subresource.

get_tags (*headers=None*)

get_versioning_status (*headers=None*)

Returns the current status of versioning configuration on the bucket.

Return type *bool*

get_website_configuration (*headers=None*)

Returns the current status of website configuration on the bucket.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type *dict*

Returns

A dictionary containing the parsed XML response from GCS. The overall structure is:

- WebsiteConfiguration
 - MainPageSuffix: suffix that is appended to request that is for a “directory” on the website endpoint.
 - NotFoundPage: name of an object to serve when site visitors encounter a 404.

get_website_configuration_obj (*headers=None*)

Get the website configuration as a `boto.s3.website.WebsiteConfiguration` object.

get_website_configuration_with_xml (*headers=None*)

Returns the current status of website configuration on the bucket as unparsed XML.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type *2-Tuple*

Returns

2-tuple containing:

1. A dictionary containing the parsed XML response from GCS. The

overall structure is:

- WebsiteConfiguration
 - MainPageSuffix: suffix that is appended to request that is for a “directory” on the website endpoint.
 - NotFoundPage: name of an object to serve when site visitors encounter a 404

2. Unparsed XML describing the bucket's website configuration.

get_website_configuration_xml (*headers=None*)

Get raw website configuration xml

get_website_endpoint ()

Returns the fully qualified hostname to use if you want to access this bucket as a website. This doesn't validate whether the bucket has been correctly configured as a website or not.

get_xml_acl (*key_name=""*, *headers=None*, *version_id=None*, *generation=None*)

Returns the ACL string of the bucket or an object in the bucket.

Parameters

- **key_name** (*str*) – The name of the object to get the ACL for. If not specified, the ACL for the bucket will be returned.
- **headers** (*dict*) – Additional headers to set during the request.
- **version_id** (*string*) – Unused in this subclass.
- **generation** (*int*) – If specified, gets the ACL for a specific generation of a versioned object. If not specified, the current version is returned. This parameter is only valid when retrieving the ACL of an object, not a bucket.

Return type *str*

get_xml_tags (*headers=None*)

initiate_multipart_upload (*key_name*, *headers=None*, *reduced_redundancy=False*, *metadata=None*, *encrypt_key=False*, *policy=None*)

Start a multipart upload operation.

Note: Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Parameters

- **key_name** (*string*) – The name of the key that will ultimately result from this multipart upload operation. This will be exactly as the key appears in the bucket after the upload process has been completed.
- **headers** (*dict*) – Additional HTTP headers to send and store with the resulting key in S3.
- **reduced_redundancy** (*boolean*) – In multipart uploads, the storage class is specified when initiating the upload, not when uploading individual parts. So if you want the resulting key to use the reduced redundancy storage class set this flag when you initiate the upload.
- **metadata** (*dict*) – Any metadata that you would like to set on the key that results from the multipart upload.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **policy** (*boto.s3.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key (once completed) in S3.

list (*prefix=""*, *delimiter=""*, *marker=""*, *headers=None*, *encoding_type=None*)

List key objects within a bucket. This returns an instance of an `BucketListResultSet` that automatically handles all of the result paging, etc. from S3. You just need to keep iterating until there are no more results.

Called with no arguments, this will return an iterator object across all keys within the bucket.

The Key objects returned by the iterator are obtained by parsing the results of a GET on the bucket, also known as the List Objects request. The XML returned by this request contains only a subset of the information about each key. Certain metadata fields such as Content-Type and user metadata are not available in the XML. Therefore, if you want these additional metadata fields you will have to do a HEAD request on the Key in the bucket.

Parameters

- **prefix** (*string*) – allows you to limit the listing to a particular prefix. For example, if you call the method with `prefix='/foo/'` then the iterator will only cycle through the keys that begin with the string `'/foo/'`.
- **delimiter** (*string*) – can be used in conjunction with the prefix to allow you to organize and browse your keys hierarchically. See <http://goo.gl/Xx63h> for more details.
- **marker** (*string*) – The “marker” of where you are in the result set
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

Return type `boto.s3.bucketlistresultset.BucketListResultSet`

Returns an instance of a `BucketListResultSet` that handles paging, etc

list_grants (*headers=None*)

Returns the ACL entries applied to this bucket.

Parameters **headers** (*dict*) – Additional headers to send with the request.

Return type list containing `Entry` objects.

list_multipart_uploads (*key_marker=""*, *upload_id_marker=""*, *headers=None*, *encoding_type=None*)

List multipart upload objects within a bucket. This returns an instance of an `MultiPartUploadListResultSet` that automatically handles all of the result paging, etc. from S3. You just need to keep iterating until there are no more results.

Parameters

- **key_marker** (*string*) – The “marker” of where you are in the result set
- **upload_id_marker** (*string*) – The upload identifier
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: url

Return type `boto.s3.bucketlistresultset.MultiPartUploadListResultSet`

Returns an instance of a BucketListResultSet that handles paging, etc

list_versions (*prefix=""*, *delimiter=""*, *marker=""*, *generation_marker=""*, *headers=None*)

List versioned objects within a bucket. This returns an instance of an VersionedBucketListResultSet that automatically handles all of the result paging, etc. from GCS. You just need to keep iterating until there are no more results. Called with no arguments, this will return an iterator object across all keys within the bucket.

Parameters

- **prefix** (*string*) – allows you to limit the listing to a particular prefix. For example, if you call the method with `prefix='/foo/'` then the iterator will only cycle through the keys that begin with the string `'/foo/'`.
- **delimiter** (*string*) – can be used in conjunction with the prefix to allow you to organize and browse your keys hierarchically. See: <https://developers.google.com/storage/docs/reference-headers#delimiter> for more details.
- **marker** (*string*) – The “marker” of where you are in the result set
- **generation_marker** (*string*) – The “generation marker” of where you are in the result set.
- **headers** (*dict*) – A dictionary of header name/value pairs.

Return type `boto.gs.bucketlistresultset.VersionedBucketListResultSet`

Returns an instance of a BucketListResultSet that handles paging, etc.

lookup (*key_name*, *headers=None*)

Deprecated: Please use `get_key` method.

Parameters **key_name** (*string*) – The name of the key to retrieve

Return type `boto.s3.key.Key`

Returns A Key object from this bucket.

make_public (*recursive=False*, *headers=None*)

new_key (*key_name=None*)

Creates a new key

Parameters **key_name** (*string*) – The name of the key to create

Return type `boto.s3.key.Key` or subclass

Returns An instance of the newly created key object

set_acl (*acl_or_str*, *key_name=""*, *headers=None*, *version_id=None*, *generation=None*, *if_generation=None*, *if_metageneration=None*)

Sets or changes a bucket's or key's ACL.

Parameters

- **acl_or_str** (*string* or `boto.gs.acl.ACL`) – A canned ACL string (see `CannedACLStrings`) or an ACL object.
- **key_name** (*string*) – A key name within the bucket to set the ACL for. If not specified, the ACL for the bucket will be set.
- **headers** (*dict*) – Additional headers to set during the request.
- **version_id** (*string*) – Unused in this subclass.

- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

set_as_logging_target (*headers=None*)

Setup the current bucket as a logging target by granting the necessary permissions to the LogDelivery group to write log files to this bucket.

set_canned_acl (*acl_str, key_name="", headers=None, version_id=None, generation=None, if_generation=None, if_metageneration=None*)

Sets a bucket's or objects's ACL using a predefined (canned) value.

Parameters

- **acl_str** (*string*) – A canned ACL string. See *CannedACLStrings*.
- **key_name** (*string*) – A key name within the bucket to set the ACL for. If not specified, the ACL for the bucket will be set.
- **headers** (*dict*) – Additional headers to set during the request.
- **version_id** (*string*) – Unused in this subclass.
- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

set_cors (*cors, headers=None*)

Sets a bucket's CORS XML document.

Parameters

- **cors** (*str*) – A string containing the CORS XML.
- **headers** (*dict*) – Additional headers to send with the request.

set_cors_xml (*cors_xml, headers=None*)

Set the CORS (Cross-Origin Resource Sharing) for a bucket.

Parameters **cors_xml** (*str*) – The XML document describing your desired CORS configuration. See the S3 documentation for details of the exact syntax required.

set_def_acl (*acl_or_str, headers=None*)

Sets or changes a bucket's default ACL.

Parameters

- **acl_or_str** (*string* or *boto.gs.acl.ACL*) – A canned ACL string (see *CannedACLStrings*) or an ACL object.
- **headers** (*dict*) – Additional headers to set during the request.

set_def_canned_acl (*acl_str, headers=None*)

Sets a bucket's default ACL using a predefined (canned) value.

Parameters

- **acl_str** (*string*) – A canned ACL string. See *CannedACLStrings*.
- **headers** (*dict*) – Additional headers to set during the request.

set_def_xml_acl (*acl_str*, *headers=None*)

Sets a bucket's default ACL to an XML string.

Parameters

- **acl_str** (*string*) – A string containing the ACL XML.
- **headers** (*dict*) – Additional headers to set during the request.

set_encryption_config (*default_kms_key_name=None*, *headers=None*)

Sets a bucket's EncryptionConfig XML document.

Parameters

- **default_kms_key_name** (*str*) – A string containing a fully-qualified Cloud KMS key name.
- **headers** (*dict*) – Additional headers to send with the request.

set_key_class (*key_class*)

Set the Key class associated with this bucket. By default, this would be the boto.s3.key.Key class but if you want to subclass that for some reason this allows you to associate your new class with a bucket so that when you call bucket.new_key() or when you get a listing of keys in the bucket you will get an instances of your key class rather than the default.

Parameters **key_class** (*class*) – A subclass of Key that can be more specific

set_policy (*policy*, *headers=None*)

Add or replace the JSON policy associated with the bucket.

Parameters **policy** (*str*) – The JSON policy as a string.

set_request_payment (*payer='BucketOwner'*, *headers=None*)

set_storage_class (*storage_class*, *headers=None*)

Sets a bucket's storage class.

Parameters

- **storage_class** (*str*) – A string containing the storage class.
- **headers** (*dict*) – Additional headers to send with the request.

set_subresource (*subresource*, *value*, *key_name=""*, *headers=None*, *version_id=None*)

Set a subresource for a bucket or key.

Parameters

- **subresource** (*string*) – The subresource to set.
- **value** (*string*) – The value of the subresource.
- **key_name** (*string*) – The key to operate on, or None to operate on the bucket.
- **headers** (*dict*) – Additional HTTP headers to include in the request.
- **src_version_id** (*string*) – Optional. The version id of the key to operate on. If not specified, operate on the newest version.

set_tags (*tags*, *headers=None*)

set_website_configuration (*config*, *headers=None*)

Parameters **config** (*boto.s3.website.WebsiteConfiguration*) – Configuration data

set_website_configuration_xml (*xml*, *headers=None*)

Upload xml website configuration

set_xml_acl (*acl_str*, *key_name=""*, *headers=None*, *version_id=None*, *query_args='acl'*, *generation=None*, *if_generation=None*, *if_metageneration=None*)

Sets a bucket's or objects's ACL to an XML string.

Parameters

- **acl_str** (*string*) – A string containing the ACL XML.
- **key_name** (*string*) – A key name within the bucket to set the ACL for. If not specified, the ACL for the bucket will be set.
- **headers** (*dict*) – Additional headers to set during the request.
- **version_id** (*string*) – Unused in this subclass.
- **query_args** (*str*) – The query parameters to pass with the request.
- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

set_xml_logging (*logging_str*, *headers=None*)

Set logging on a bucket directly to the given xml string.

Parameters **logging_str** (*unicode string*) – The XML for the bucket logging status which will be set. The string will be converted to utf-8 before it is sent. Usually, you will obtain this XML from the BucketLogging object.

Return type *bool*

Returns True if ok or raises an exception.

set_xml_tags (*tag_str*, *headers=None*, *query_args='tagging'*)

validate_get_all_versions_params (*params*)

See documentation in boto/s3/bucket.py.

validate_kwarg_names (*kwargs*, *names*)

Checks that all named arguments are in the specified list of names.

Parameters

- **kwargs** (*dict*) – Dictionary of kwargs to validate.
- **names** (*list*) – List of possible named arguments.

4.136.3 boto.gs.bucketlistresultset

```
class boto.gs.bucketlistresultset.VersionedBucketListResultSet (bucket=None,
                                                             prefix="", de-
                                                             limiter="",
                                                             marker="",
                                                             genera-
                                                             tion_marker="",
                                                             headers=None)
```

A resultset for listing versions within a bucket. Uses the bucket_lister generator function and implements the iterator interface. This transparently handles the results paging from GCS so even if you have many thousands of keys within the bucket you can iterate over all keys in a reasonably efficient manner.

```
boto.gs.bucketlistresultset.versioned_bucket_lister (bucket, prefix="", delim-
                                                         iter="", marker="", genera-
                                                         tion_marker="", headers=None)
```

A generator function for listing versioned objects.

4.136.4 boto.gs.connection

```

class boto.gs.connection.GSConnection (gs_access_key_id=None,
                                       gs_secret_access_key=None,      is_secure=True,
                                       port=None, proxy=None, proxy_port=None,
                                       proxy_user=None, proxy_pass=None,
                                       host='storage.googleapis.com', debug=0,
                                       https_connection_factory=None, calling_format=<boto.s3.connection.SubdomainCallingFormat
                                       object>, path='/', suppress_consec_slashes=True)

DefaultCallingFormat = 'boto.s3.connection.SubdomainCallingFormat'
DefaultHost = 'storage.googleapis.com'
QueryString = 'Signature=%s&Expires=%d&GoogleAccessId=%s'
access_key
auth_region_name
auth_service_name
aws_access_key_id
aws_secret_access_key
build_base_http_request (method, path, auth_path, params=None, headers=None, data="",
                        host=None)
build_post_form_args (bucket_name, key, expires_in=6000, acl=None, success_action_redirect=None,
                    max_content_length=None, http_method='http', fields=None, conditions=None, storage_class='STANDARD',
                    server_side_encryption=None)

```

Taken from the AWS book Python examples and modified for use with boto This only returns the arguments required for the post form, not the actual form. This does not return the file input field which also needs to be added

Parameters

- **bucket_name** (*string*) – Bucket to submit to
- **key** (*string*) – Key name, optionally add `/${filename}` to the end to attach the submitted filename
- **expires_in** (*integer*) – Time (in seconds) before this expires, defaults to 6000
- **acl** (*string*) – A canned ACL. One of: `* private * public-read * public-read-write * authenticated-read * bucket-owner-read * bucket-owner-full-control`
- **success_action_redirect** (*string*) – URL to redirect to on success
- **max_content_length** (*integer*) – Maximum size for this file
- **http_method** (*string*) – HTTP Method to use, “http” or “https”
- **storage_class** (*string*) – Storage class to use for storing the object. Valid values: `STANDARD | REDUCED_REDUNDANCY`
- **server_side_encryption** (*string*) – Specifies server-side encryption algorithm to use when Amazon S3 creates an object. Valid values: `None | AES256`

Return type dict

Returns

A dictionary containing field names/values as well as a url to POST to

build_post_policy (*expiration_time, conditions*)

Taken from the AWS book Python examples and modified for use with boto

close ()

(Optional) Close any open HTTP connections. This is non-destructive, and making a new request will open a connection again.

connection

create_bucket (*bucket_name, headers=None, location='US', policy=None, storage_class='STANDARD'*)

Creates a new bucket. By default it's located in the USA. You can pass Location.EU to create bucket in the EU. You can also pass a LocationConstraint for where the bucket should be located, and a StorageClass describing how the data should be stored.

Parameters

- **bucket_name** (*string*) – The name of the new bucket.
- **headers** (*dict*) – Additional headers to pass along with the request to GCS.
- **location** (*boto.gs.connection.Location*) – The location of the new bucket.
- **policy** (*boto.gs.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key in GCS.
- **storage_class** (*string*) – Either 'STANDARD' or 'DURABLE_REDUCED_AVAILABILITY'.

delete_bucket (*bucket, headers=None*)

Removes an S3 bucket.

In order to remove the bucket, it must first be empty. If the bucket is not empty, an S3ResponseError will be raised.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.

generate_url (*expires_in, method, bucket="", key="", headers=None, query_auth=True, force_http=False, response_headers=None, expires_in_absolute=False, version_id=None*)

generate_url_sigv4 (*expires_in, method, bucket="", key="", headers=None, force_http=False, response_headers=None, version_id=None, iso_date=None*)

get_all_buckets (*headers=None*)

get_bucket (*bucket_name, validate=True, headers=None*)

Retrieves a bucket by name.

If the bucket does not exist, an S3ResponseError will be raised. If you are unsure if the bucket exists or not, you can use the S3Connection.lookup method, which will either return a valid bucket or None.

Parameters

- **bucket_name** (*string*) – The name of the bucket

- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **validate** (*boolean*) – If True, it will try to fetch all keys within the given bucket. (Default: True)

get_canonical_user_id (*headers=None*)

Convenience method that returns the “CanonicalUserID” of the user who’s credentials are associated with the connection. The only way to get this value is to do a GET request on the service which returns all buckets associated with the account. As part of that response, the canonical userid is returned. This method simply does all of that and then returns just the user id.

Return type string

Returns A string containing the canonical user id.

get_http_connection (*host, port, is_secure*)

get_path (*path='/'*)

get_proxy_auth_header ()

get_proxy_url_with_auth ()

gs_access_key_id

gs_secret_access_key

handle_proxy (*proxy, proxy_port, proxy_user, proxy_pass*)

head_bucket (*bucket_name, headers=None*)

Determines if a bucket exists by name.

If the bucket does not exist, an S3ResponseError will be raised.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.

Returns A <Bucket> object

lookup (*bucket_name, validate=True, headers=None*)

Attempts to get a bucket from S3.

Works identically to S3Connection.get_bucket, save for that it will return None if the bucket does not exist instead of throwing an exception.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **validate** (*boolean*) – If True, it will try to fetch all keys within the given bucket. (Default: True)

make_request (*method, bucket=”, key=”, headers=None, data=”, query_args=None, sender=None, override_num_retries=None, retry_handler=None*)

Makes a request to the server, with stock multiple-retry logic.

new_http_connection (*host, port, is_secure*)

prefix_proxy_to_path (*path, host=None*)

profile_name

proxy_ssl (*host=None, port=None*)

put_http_connection (*host, port, is_secure, connection*)

secret_key

server_name (*port=None*)

set_bucket_class (*bucket_class*)

Set the Bucket class associated with this bucket. By default, this would be the `boto.s3.key.Bucket` class but if you want to subclass that for some reason this allows you to associate your new class.

Parameters **bucket_class** (*class*) – A subclass of `Bucket` that can be more specific

set_host_header (*request*)

set_request_hook (*hook*)

skip_proxy (*host*)

class `boto.gs.connection.Location`

DEFAULT = 'US'

EU = 'EU'

4.136.5 boto.gs.cors

class `boto.gs.cors.Cors`

Encapsulates the CORS configuration XML document

endElement (*name, value, connection*)

SAX XML logic for parsing new element found.

startElement (*name, attrs, connection*)

SAX XML logic for parsing new element found.

to_xml ()

Convert CORS object into XML string representation.

validateParseLevel (*tag, level*)

Verify parse level for a given tag.

4.136.6 boto.gs.key

class `boto.gs.key.Key` (*bucket=None, name=None, generation=None*)

Represents a key (object) in a GS bucket.

Variables

- **bucket** – The parent `boto.gs.bucket.Bucket`.
- **name** – The name of this Key object.
- **metadata** – A dictionary containing user metadata that you wish to store with the object or that has been retrieved from an existing object.
- **cache_control** – The value of the *Cache-Control* HTTP header.
- **content_type** – The value of the *Content-Type* HTTP header.
- **content_encoding** – The value of the *Content-Encoding* HTTP header.
- **content_disposition** – The value of the *Content-Disposition* HTTP header.
- **content_language** – The value of the *Content-Language* HTTP header.
- **etag** – The *etag* associated with this object.

- **last_modified** – The string timestamp representing the last time this object was modified in GS.
- **owner** – The ID of the owner of this object.
- **storage_class** – The storage class of the object. Currently, one of: STANDARD | DURABLE_REDUCED_AVAILABILITY.
- **md5** – The MD5 hash of the contents of the object.
- **size** – The size, in bytes, of the object.
- **generation** – The generation number of the object.
- **metageneration** – The generation number of the object metadata.
- **encrypted** – Whether the object is encrypted while at rest on the server.
- **cloud_hashes** – Dictionary of checksums as supplied by the storage provider.

BufferSize = 8192

DefaultContentType = 'application/octet-stream'

RestoreBody = '<?xml version="1.0" encoding="UTF-8"?>\n <RestoreRequest xmlns="http://'

add_email_grant (*permission, email_address*)

Convenience method that provides a quick way to add an email grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: READIFULL_CONTROL See <http://code.google.com/apis/storage/docs/developer-guide.html#authorization> for more details on permissions.
- **email_address** (*string*) – The email address associated with the Google account to which you are granting the permission.

add_group_email_grant (*permission, email_address, headers=None*)

Convenience method that provides a quick way to add an email group grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: READIFULL_CONTROL See <http://code.google.com/apis/storage/docs/developer-guide.html#authorization> for more details on permissions.
- **email_address** (*string*) – The email address associated with the Google Group to which you are granting the permission.

add_group_grant (*permission, group_id*)

Convenience method that provides a quick way to add a canonical group grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: READIFULL_CONTROL See <http://code.google.com/apis/storage/docs/developer-guide.html#authorization> for more details on permissions.
- **group_id** (*string*) – The canonical group id associated with the Google Groups account you are granting the permission to.

add_user_grant (*permission, user_id*)

Convenience method that provides a quick way to add a canonical user grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to GS.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: READ|FULL_CONTROL See <http://code.google.com/apis/storage/docs/developer-guide.html#authorization> for more details on permissions.
- **user_id** (*string*) – The canonical user id associated with the GS account to which you are granting the permission.

base64md5

base_fields = set(['content-length', 'content-language', 'content-disposition', 'content-encoding', 'content-type', 'etag', 'last-modified', 'location', 'range'])

base_user_settable_fields = set(['content-disposition', 'content-language', 'content-encoding', 'content-type', 'etag', 'last-modified', 'location', 'range'])

change_storage_class (*new_storage_class, dst_bucket=None, validate_dst_bucket=True*)

Change the storage class of an existing key. Depending on whether a different destination bucket is supplied or not, this will either move the item within the bucket, preserving all metadata and ACL info bucket changing the storage class or it will copy the item to the provided destination bucket, also preserving metadata and ACL info.

Parameters

- **new_storage_class** (*string*) – The new storage class for the Key. Possible values are: * STANDARD * REDUCED_REDUNDANCY
- **dst_bucket** (*string*) – The name of a destination bucket. If not provided the current bucket of the key will be used.
- **validate_dst_bucket** (*bool*) – If True, will validate the dst_bucket by using an extra list request.

close (*fast=False*)

Close this key.

Parameters **fast** (*bool*) – True if you want the connection to be closed without first

reading the content. This should only be used in cases where subsequent calls don't need to return the content from the open HTTP connection. Note: As explained at <http://docs.python.org/2/library/httplib.html#httplib.HTTPConnection.getresponse>, callers must read the whole response before sending a new request to the server. Calling Key.close(fast=True) and making a subsequent request to the server will work because boto will get an httplib exception and close/reopen the connection.

closed = False**compose** (*components, content_type=None, headers=None*)

Create a new object from a sequence of existing objects.

The content of the object representing this Key will be the concatenation of the given object sequence. For more detail, visit

<https://developers.google.com/storage/docs/composite-objects>

:type components list of Keys :param components List of gs.Keys representing the component objects

:type content_type (optional) string :param content_type Content type for the new composite object.

compute_hash (*fp, algorithm, size=None*)**Parameters**

- **fp** (*file*) – File pointer to the file to hash. The file pointer will be reset to the same position before the method returns.

- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where the file is being split in place into different parts. Less bytes may be available.

compute_md5 (*fp*, *size=None*)

Parameters

- **fp** (*file*) – File pointer to the file to MD5 hash. The file pointer will be reset to the same position before the method returns.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where the file is being split in place into different parts. Less bytes may be available.

copy (*dst_bucket*, *dst_key*, *metadata=None*, *reduced_redundancy=False*, *preserve_acl=False*, *encrypt_key=False*, *validate_dst_bucket=True*)

Copy this Key to another bucket.

Parameters

- **dst_bucket** (*string*) – The name of the destination bucket
- **dst_key** (*string*) – The name of the destination key
- **metadata** (*dict*) – Metadata to be associated with new key. If metadata is supplied, it will replace the metadata of the source key being copied. If no metadata is supplied, the source key's metadata will be copied to the new key.
- **reduced_redundancy** (*bool*) – If True, this will force the storage class of the new Key to be REDUCED_REDUNDANCY regardless of the storage class of the key being copied. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **preserve_acl** (*bool*) – If True, the ACL from the source key will be copied to the destination key. If False, the destination key will have the default ACL. Note that preserving the ACL in the new key object will require two additional API calls to S3, one to retrieve the current ACL and one to set that ACL on the new object. If you don't care about the ACL, a value of False will be significantly more efficient.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **validate_dst_bucket** (*bool*) – If True, will validate the *dst_bucket* by using an extra list request.

Return type *boto.s3.key.Key* or subclass

Returns An instance of the newly created key object

delete (*headers=None*)

Delete this key from S3

endElement (*name*, *value*, *connection*)

exists (*headers=None*)

Returns True if the key exists

Return type *bool*

Returns Whether the key exists on S3

f = 'x-robots-tag'

generate_url (*expires_in*, *method*='GET', *headers*=None, *query_auth*=True, *force_http*=False, *response_headers*=None, *expires_in_absolute*=False, *version_id*=None, *policy*=None, *reduced_redundancy*=False, *encrypt_key*=False)

Generate a URL to access this key.

Parameters

- **expires_in** (*int*) – How long the url is valid for, in seconds.
- **method** (*string*) – The method to use for retrieving the file (default is GET).
- **headers** (*dict*) – Any headers to pass along in the request.
- **query_auth** (*bool*) – If True, signs the request in the URL.
- **force_http** (*bool*) – If True, http will be used instead of https.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **expires_in_absolute** (*bool*) –
- **version_id** (*string*) – The version_id of the object to GET. If specified this overrides any value in the key.
- **policy** (`boto.s3.acl.CannedACLStrings`) – A canned ACL policy that will be applied to the new key in S3.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.

Return type string

Returns The URL to access the key

get_acl (*headers*=None, *generation*=None)

Returns the ACL of this object.

Parameters

- **headers** (*dict*) – Additional headers to set during the request.
- **generation** (*int*) – If specified, gets the ACL for a specific generation of a versioned object. If not specified, the current version is returned.

Return type `gs.acl.ACL`

get_contents_as_string (*headers*=None, *cb*=None, *num_cb*=10, *torrent*=False, *version_id*=None, *response_headers*=None, *encoding*=None)

Retrieve an object from S3 using the name of the Key object as the key in S3. Return the contents of the object as a string. See `get_contents_to_file` method for details about the parameters.

Parameters

- **headers** (*dict*) – Any additional headers to send in the request
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.

- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.
- **encoding** (*str*) – The text encoding to use, such as `utf-8` or `iso-8859-1`. If set, then a string will be returned. Defaults to `None` and returns bytes.

Return type `bytes` or `str`

Returns The contents of the file as bytes or a string

get_contents_to_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *res_download_handler=None*, *response_headers=None*, *hash_algs=None*)

Retrieve an object from GCS using the name of the Key object as the key in GCS. Write the contents of the object to the file pointed to by 'fp'.

Parameters

- **fp** (*File -like object*) –
- **headers** (*dict*) – additional HTTP headers that will be sent with the GET request.
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GCS and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **res_download_handler** – If provided, this handler will perform the download.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/sMkcC> for details.

get_contents_to_filename (*filename*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *res_download_handler=None*, *response_headers=None*)

Retrieve an object from S3 using the name of the Key object as the key in S3. Store contents of the object to a file named by 'filename'. See `get_contents_to_file` method for details about the parameters.

Parameters

- **filename** (*string*) – The filename of where to put the file contents
- **headers** (*dict*) – Any additional headers to send in the request

- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** (*int*) – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **res_download_handler** – If provided, this handler will perform the download.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.

get_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *override_num_retries=None*, *response_headers=None*, *hash_algs=None*)

Retrieves a file from an S3 Key

Parameters

- **fp** (*file*) – File pointer to put the data into
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – Flag for whether to get a torrent for the file
- **override_num_retries** (*int*) – If not `None` will override configured `num_retries` parameter for underlying GET.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.

Param headers to send when retrieving the files

get_md5_from_hexdigest (*md5_hexdigest*)

A utility function to create the 2-tuple (`md5hexdigest`, `base64md5`) from just having a precalculated `md5_hexdigest`.

get_metadata (*name*)

get_redirect ()

Return the redirect location configured for this key.

If no redirect is configured (via `set_redirect`), then `None` will be returned.

get_torrent_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*)

Get a torrent file (see to `get_file`)

Parameters

- **fp** (*file*) – The file pointer of where to put the torrent
- **headers** (*dict*) – Headers to be passed
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.

get_xml_acl (*headers=None*, *generation=None*)

Returns the ACL string of this object.

Parameters

- **headers** (*dict*) – Additional headers to set during the request.
- **generation** (*int*) – If specified, gets the ACL for a specific generation of a versioned object. If not specified, the current version is returned.

Return type *str*

handle_addl_headers (*headers*)

Used by Key subclasses to do additional, provider-specific processing of response headers. No-op for this base class.

handle_encryption_headers (*resp*)

handle_restore_headers (*response*)

handle_storage_class_header (*resp*)

handle_version_headers (*resp*, *force=False*)

key

make_public (*headers=None*)

md5

next ()

By providing a next method, the key object supports use as an iterator. For example, you can now say:

for bytes in key: write bytes to a file or whatever

All of the HTTP connection stuff is handled for you.

open (*mode='r'*, *headers=None*, *query_args=None*, *override_num_retries=None*)

open_read (*headers=None*, *query_args=""*, *override_num_retries=None*, *response_headers=None*)

Open this key for reading

Parameters

- **headers** (*dict*) – Headers to pass in the web request
- **query_args** (*string*) – Arguments to pass in the query string (ie, 'torrent')

- **override_num_retries** (*int*) – If not None will override configured num_retries parameter for underlying GET.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.

open_write (*headers=None, override_num_retries=None*)

Open this key for writing. Not yet implemented

Parameters

- **headers** (*dict*) – Headers to pass in the write request
- **override_num_retries** (*int*) – If not None will override configured num_retries parameter for underlying PUT.

provider

read (*size=0*)

restore (*days, headers=None*)

Restore an object from an archive.

Parameters **days** (*int*) – The lifetime of the restored object (must be at least 1 day). If the object is already restored then this parameter can be used to readjust the lifetime of the restored object. In this case, the days param is with respect to the initial time of the request. If the object has not been restored, this param is with respect to the completion time of the request.

send_file (*fp, headers=None, cb=None, num_cb=10, query_args=None, chunked_transfer=False, size=None, hash_algs=None*)

Upload a file to GCS.

Parameters

- **fp** (*file*) – The file pointer to upload. The file pointer must point at the offset from which you wish to upload. ie. if uploading the full file, it should point at the start of the file. Normally when a file is opened for reading, the fp will point at the first byte. See the bytes parameter below for more info.
- **headers** (*dict*) – The headers to pass along with the PUT request
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer. Providing a negative integer will cause your callback to be called with each buffer read.
- **query_args** (*string*) – Arguments to pass in the query string.
- **chunked_transfer** (*boolean*) – (optional) If true, we use chunked Transfer-Encoding.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the default behaviour is to read all bytes from the file pointer. Less bytes may be available.
- **hash_algs** (*dictionary*) – (optional) Dictionary of hash algorithms and corresponding hashing class that implements update() and digest(). Defaults to {'md5': hashlib.md5}.

set_acl (*acl_or_str*, *headers=None*, *generation=None*, *if_generation=None*,
if_metageneration=None)
Sets the ACL for this object.

Parameters

- **acl_or_str** (*string* or *boto.gs.acl.ACL*) – A canned ACL string (see *CannedACLStrings*) or an ACL object.
- **headers** (*dict*) – Additional headers to set during the request.
- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

set_canned_acl (*acl_str*, *headers=None*, *generation=None*, *if_generation=None*,
if_metageneration=None)
Sets this object's ACL using a predefined (canned) value.

Parameters

- **acl_str** (*string*) – A canned ACL string. See *CannedACLStrings*.
- **headers** (*dict*) – Additional headers to set during the request.
- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

set_contents_from_file (*fp*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*,
md5=None, *res_upload_handler=None*, *size=None*,
rewind=False, *if_generation=None*)

Store an object in GS using the name of the Key object as the key in GS and the contents of the file pointed to by 'fp' as the contents.

Parameters

- **fp** (*file*) – The file whose contents are to be uploaded.
- **headers** (*dict*) – (optional) Additional HTTP headers to be sent with the PUT request.
- **replace** (*bool*) – (optional) If this parameter is False, the method will first check to see if an object exists in the bucket with the same key. If it does, it won't overwrite it. The default value is True which will overwrite the object.
- **cb** (*function*) – (optional) Callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GS and the second representing the total number of bytes that need to be transmitted.
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter, this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.

- **policy** (*boto.gs.acl.CannedACLStrings*) – (optional) A canned ACL policy that will be applied to the new key in GS.
- **md5** (*tuple*) – (optional) A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.

If you need to compute the MD5 for any reason prior to upload, it's silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.

- **res_upload_handler** (*boto.gs.resumable_upload_handler.ResumableUploadHandler*) – (optional) If provided, this handler will perform the upload.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the default behaviour is to read all bytes from the file pointer. Less bytes may be available.

Notes:

1. The “size” parameter currently cannot be used when a resumable upload handler is given but is still useful for uploading part of a file as implemented by the parent class.
 2. At present Google Cloud Storage does not support multipart uploads.
- **rewind** (*bool*) – (optional) If True, the file pointer (fp) will be rewound to the start before any bytes are read from it. The default behaviour is False which reads from the current position of the file pointer (fp).
 - **if_generation** (*int*) – (optional) If set to a generation number, the object will only be written to if its current generation number is this value. If set to the value 0, the object will only be written if it doesn't already exist.

Return type `int`

Returns The number of bytes written to the key.

TODO: At some point we should refactor the Bucket and Key classes, to move functionality common to all providers into a parent class, and provider-specific functionality into subclasses (rather than just overriding/sharing code the way it currently works).

set_contents_from_filename (*filename, headers=None, replace=True, cb=None, num_cb=10, policy=None, md5=None, reduced_redundancy=None, res_upload_handler=None, if_generation=None*)

Store an object in GS using the name of the Key object as the key in GS and the contents of the file named by 'filename'. See `set_contents_from_file` method for details about the parameters.

Parameters

- **filename** (*string*) – The name of the file that you want to put onto GS.
- **headers** (*dict*) – (optional) Additional headers to pass along with the request to GS.
- **replace** (*bool*) – (optional) If True, replaces the contents of the file if it already exists.
- **cb** (*function*) – (optional) Callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the

first representing the number of bytes that have been successfully transmitted to GS and the second representing the total number of bytes that need to be transmitted.

- **num_cb** (*int*) – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (**:py:attribute:'boto.gs.acl.CannedACLStrings'**) – (optional) A canned ACL policy that will be applied to the new key in GS.
- **md5** (*tuple*) – (optional) A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.

If you need to compute the MD5 for any reason prior to upload, it's silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.

- **res_upload_handler** (*boto.gs.resumable_upload_handler.ResumableUploadHandler*) – (optional) If provided, this handler will perform the upload.
- **if_generation** (*int*) – (optional) If set to a generation number, the object will only be written to if its current generation number is this value. If set to the value 0, the object will only be written if it doesn't already exist.

set_contents_from_stream (*args, **kwargs)

Store an object using the name of the Key object as the key in cloud and the contents of the data stream pointed to by 'fp' as the contents.

The stream object is not seekable and total size is not known. This has the implication that we can't specify the Content-Size and Content-MD5 in the header. So for huge uploads, the delay in calculating MD5 is avoided but with a penalty of inability to verify the integrity of the uploaded data.

Parameters

- **fp** (*file*) – the file whose contents are to be uploaded
- **headers** (*dict*) – additional HTTP headers to be sent with the PUT request.
- **replace** (*bool*) – If this parameter is False, the method will first check to see if an object exists in the bucket with the same key. If it does, it won't overwrite it. The default value is True which will overwrite the object.
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GS and the second representing the total number of bytes that need to be transmitted.
- **num_cb** (*int*) – (optional) If a callback is specified with the `cb` parameter, this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (*boto.gs.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key in GS.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the

default behaviour is to read all bytes from the file pointer. Less bytes may be available.

- **if_generation** (*int*) – (optional) If set to a generation number, the object will only be written to if its current generation number is this value. If set to the value 0, the object will only be written if it doesn't already exist.

set_contents_from_string (*s*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*, *md5=None*, *if_generation=None*)

Store an object in GCS using the name of the Key object as the key in GCS and the string 's' as the contents. See `set_contents_from_file` method for details about the parameters.

Parameters

- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **replace** (*bool*) – If True, replaces the contents of the file if it already exists.
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GCS and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (*boto.gs.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key in GCS.
- **md5** (A *tuple* containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.) – If you need to compute the MD5 for any reason prior to upload, it's silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.
- **if_generation** (*int*) – (optional) If set to a generation number, the object will only be written to if its current generation number is this value. If set to the value 0, the object will only be written if it doesn't already exist.

set_metadata (*name*, *value*)

set_redirect (*redirect_location*, *headers=None*)

Configure this key to redirect to another location.

When the bucket associated with this key is accessed from the website endpoint, a 301 redirect will be issued to the specified *redirect_location*.

Parameters **redirect_location** (*string*) – The location to redirect.

set_remote_metadata (*metadata_plus*, *metadata_minus*, *preserve_acl*, *headers=None*)

set_xml_acl (*acl_str*, *headers=None*, *generation=None*, *if_generation=None*, *if_metageneration=None*)

Sets this object's ACL to an XML string.

Parameters

- **acl_str** (*string*) – A string containing the ACL XML.
- **headers** (*dict*) – Additional headers to set during the request.

- **generation** (*int*) – If specified, sets the ACL for a specific generation of a versioned object. If not specified, the current version is modified.
- **if_generation** (*int*) – (optional) If set to a generation number, the acl will only be updated if its current generation number is this value.
- **if_metageneration** (*int*) – (optional) If set to a metageneration number, the acl will only be updated if its current metageneration number is this value.

should_retry (*response, chunked_transfer=False*)

startElement (*name, attrs, connection*)

storage_class

update_metadata (*d*)

4.136.7 boto.gs.user

class boto.gs.user.**User** (*parent=None, id="", name=""*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml (*element_name='Owner'*)

4.136.8 boto.gs.resumable_upload_handler

class boto.gs.resumable_upload_handler.**ResumableUploadHandler** (*tracker_file_name=None, num_retries=None*)

Constructor. Instantiate once for each uploaded file.

Parameters

- **tracker_file_name** (*string*) – optional file name to save tracker URI. If supplied and the current process fails the upload, it can be retried in a new process. If called with an existing file containing a valid tracker URI, we'll resume the upload from this URI; else we'll start a new resumable upload (and write the URI to this tracker file).
- **num_retries** (*int*) – the number of times we'll re-try a resumable upload making no progress. (Count resets every time we get progress, so upload can span many more than this number of retries.)

BUFFER_SIZE = 8192

RETRYABLE_EXCEPTIONS = (<class 'httplib.HTTPException'>, <type 'exceptions IOError'> ,

SERVER_HAS_NOTHING = (0, -1)

get_tracker_uri ()

Returns upload tracker URI, or None if the upload has not yet started.

get_upload_id ()

Returns the upload ID for the resumable upload, or None if the upload has not yet started.

handle_resumable_upload_exception (*e, debug*)

send_file (*key, fp, headers, cb=None, num_cb=10, hash_algs=None*)

Upload a file to a key into a bucket on GS, using GS resumable upload protocol.

Parameters

- **key** (*boto.s3.key.Key* or subclass) – The Key object to which data is to be uploaded
- **fp** (*file-like object*) – The file pointer to upload
- **headers** (*dict*) – The headers to pass along with the PUT request
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GS, and the second representing the total number of bytes that need to be transmitted.
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter, this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer. Providing a negative integer will cause your callback to be called with each buffer read.
- **hash_algs** (*dictionary*) – (optional) Dictionary mapping hash algorithm descriptions to corresponding state-ful hashing objects that implement update(), digest(), and copy() (e.g. hashlib.md5()). Defaults to {'md5': md5()}.

Raises ResumableUploadException if a problem occurs during the transfer.

track_progress_less_iterations (*server_had_bytes_before_attempt, roll_back_md5=True, debug=0*)

4.137 IAM

4.137.1 boto.iam

class boto.iam.IAMRegionInfo (*connection=None, name=None, endpoint=None, connection_cls=None*)

connect (***kw_params*)

Connect to this Region's endpoint. Returns an connection object pointing to the endpoint associated with this region. You may pass any of the arguments accepted by the connection class's constructor as keyword arguments and they will be passed along to the connection object.

Return type Connection object

Returns The connection to this regions endpoint

boto.iam.**connect_to_region** (*region_name, **kw_params*)

Given a valid region name, return a *boto.iam.connection.IAMConnection*.

Type str

Parameters **region_name** – The name of the region to connect to.

Return type *boto.iam.connection.IAMConnection* or None

Returns A connection to the given region, or None if an invalid region name is given

boto.iam.**regions** ()

Get all available regions for the IAM service.

Return type list

Returns A list of *boto.regioninfo.RegionInfo* instances

4.137.2 boto.iam.connection

```
class boto.iam.connection.IAMConnection(aws_access_key_id=None,  
                                         aws_secret_access_key=None, is_secure=True,  
                                         port=None, proxy=None, proxy_port=None,  
                                         proxy_user=None, proxy_pass=None,  
                                         host='iam.amazonaws.com', debug=0,  
                                         https_connection_factory=None, path='/', se-  
                                         curity_token=None, validate_certs=True, pro-  
                                         file_name=None)
```

APIVersion = '2010-05-08'

add_role_to_instance_profile (*instance_profile_name, role_name*)

Adds the specified role to the specified instance profile.

Parameters

- **instance_profile_name** (*string*) – Name of the instance profile to update.
- **role_name** (*string*) – Name of the role to add.

add_user_to_group (*group_name, user_name*)

Add a user to a group

Parameters

- **group_name** (*string*) – The name of the group
- **user_name** (*string*) – The to be added to the group.

attach_group_policy (*policy_arn, group_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to attach
- **group_name** (*string*) – Group to attach the policy to

attach_role_policy (*policy_arn, role_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to attach
- **role_name** (*string*) – Role to attach the policy to

attach_user_policy (*policy_arn, user_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to attach
- **user_name** (*string*) – User to attach the policy to

create_access_key (*user_name=None*)

Create a new AWS Secret Access Key and corresponding AWS Access Key ID for the specified user. The default status for new keys is Active

If the *user_name* is not specified, the *user_name* is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters **user_name** (*string*) – The username of the user

create_account_alias (*alias*)

Creates a new alias for the AWS account.

For more information on account id aliases, please see <http://goo.gl/ToB7G>

Parameters **alias** (*string*) – The alias to attach to the account.

create_group (*group_name*, *path*='')

Create a group.

Parameters

- **group_name** (*string*) – The name of the new group
- **path** (*string*) – The path to the group (Optional). Defaults to */*.

create_instance_profile (*instance_profile_name*, *path*=None)

Creates a new instance profile.

Parameters

- **instance_profile_name** (*string*) – Name of the instance profile to create.
- **path** (*string*) – The path to the instance profile.

create_login_profile (*user_name*, *password*)

Creates a login profile for the specified user, give the user the ability to access AWS services and the AWS Management Console.

Parameters

- **user_name** (*string*) – The name of the user
- **password** (*string*) – The new password for the user

create_policy (*policy_name*, *policy_document*, *path*='', *description*=None)

Create a policy.

Parameters **policy_name** (*string*) – The name of the new policy

:type *policy_document* string :param *policy_document*: The document of the new policy

Parameters

- **path** (*string*) – The path in which the policy will be created. Defaults to */*.
- **path** – A description of the new policy.

create_policy_version (*policy_arn*, *policy_document*, *set_as_default*=None)

Create a policy version.

Parameters **policy_arn** (*string*) – The ARN of the policy

:type *policy_document* string :param *policy_document*: The document of the new policy version

Parameters **set_as_default** (*bool*) – Sets the policy version as default Defaults to None.

create_role (*role_name*, *assume_role_policy_document*=None, *path*=None)

Creates a new role for your AWS account.

The policy grants permission to an EC2 instance to assume the role. The policy is URL-encoded according to RFC 3986. Currently, only EC2 instances can assume roles.

Parameters

- **role_name** (*string*) – Name of the role to create.
- **assume_role_policy_document** (*string* or *dict*) – The policy that grants an entity permission to assume the role.
- **path** (*string*) – The path to the role.

create_saml_provider (*saml_metadata_document*, *name*)

Creates an IAM entity to describe an identity provider (IdP) that supports SAML 2.0.

The SAML provider that you create with this operation can be used as a principal in a role's trust policy to establish a trust relationship between AWS and a SAML identity provider. You can create an IAM role

that supports Web-based single sign-on (SSO) to the AWS Management Console or one that supports API access to AWS.

When you create the SAML provider, you upload an a SAML metadata document that you get from your IdP and that includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP. This operation requires **'Signature Version 4'**. For more information, see **'Giving Console Access Using SAML'** and **'Creating Temporary Security Credentials for SAML Federation'** in the Using Temporary Credentials guide.

Parameters `saml_metadata_document` (*string*) – An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer's name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization's IdP.

For more information, see **'Creating Temporary Security Credentials for SAML Federation' in the Using Temporary Security Credentials guide.**

Parameters `name` (*string*) – The name of the provider to create.

create_user (*user_name, path='/'*)

Create a user.

Parameters

- `user_name` (*string*) – The name of the new user
- `path` (*string*) – The path in which the user will be created. Defaults to /.

create_virtual_mfa_device (*path, device_name*)

Creates a new virtual MFA device for the AWS account.

After creating the virtual MFA, use `enable-mfa-device` to attach the MFA device to an IAM user.

Parameters

- `path` (*string*) – The path for the virtual MFA device.
- `device_name` (*string*) – The name of the virtual MFA device. Used with `path` to uniquely identify a virtual MFA device.

deactivate_mfa_device (*user_name, serial_number*)

Deactivates the specified MFA device and removes it from association with the user.

Parameters

- `user_name` (*string*) – The username of the user
- `serial_number` (*string*) – The serial number which uniquely identifies the MFA device.

delete_access_key (*access_key_id, user_name=None*)

Delete an access key associated with a user.

If the `user_name` is not specified, it is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- `access_key_id` (*string*) – The ID of the access key to be deleted.
- `user_name` (*string*) – The username of the user

delete_account_alias (*alias*)

Deletes an alias for the AWS account.

For more information on account id aliases, please see <http://goo.gl/ToB7G>

Parameters `alias` (*string*) – The alias to remove from the account.

delete_account_password_policy ()

Delete the password policy currently set for the AWS account.

delete_group (*group_name*)

Delete a group. The group must not contain any Users or have any attached policies

Parameters `group_name` (*string*) – The name of the group to delete.

delete_group_policy (*group_name, policy_name*)

Deletes the specified policy document for the specified group.

Parameters

- `group_name` (*string*) – The name of the group the policy is associated with.
- `policy_name` (*string*) – The policy document to delete.

delete_instance_profile (*instance_profile_name*)

Deletes the specified instance profile. The instance profile must not have an associated role.

Parameters `instance_profile_name` (*string*) – Name of the instance profile to delete.

delete_login_profile (*user_name*)

Deletes the login profile associated with the specified user.

Parameters `user_name` (*string*) – The name of the user to delete.

delete_policy (*policy_arn*)

Delete a policy.

Parameters `policy_arn` (*string*) – The ARN of the policy to delete

delete_policy_version (*policy_arn, version_id*)

Delete a policy version.

Parameters

- `policy_arn` (*string*) – The ARN of the policy to delete a version from
- `version_id` (*string*) – The id of the version to delete

delete_role (*role_name*)

Deletes the specified role. The role must not have any policies attached.

Parameters `role_name` (*string*) – Name of the role to delete.

delete_role_policy (*role_name, policy_name*)

Deletes the specified policy associated with the specified role.

Parameters

- `role_name` (*string*) – Name of the role associated with the policy.
- `policy_name` (*string*) – Name of the policy to delete.

delete_saml_provider (*saml_provider_arn*)

Deletes a SAML provider.

Deleting the provider does not update any roles that reference the SAML provider as a principal in their trust policies. Any attempt to assume a role that references a SAML provider that has been deleted will fail. This operation requires ‘**Signature Version 4**’.

Parameters `saml_provider_arn` (*string*) – The Amazon Resource Name (ARN) of the SAML provider to delete.

delete_server_cert (*cert_name*)

Delete the specified server certificate.

Parameters `cert_name` (*string*) – The name of the server certificate you want to delete.

delete_signing_cert (*cert_id*, *user_name=None*)

Delete a signing certificate associated with a user.

If the *user_name* is not specified, it is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- **user_name** (*string*) – The username of the user
- **cert_id** (*string*) – The ID of the certificate.

delete_user (*user_name*)

Delete a user including the user's path, GUID and ARN.

If the *user_name* is not specified, the *user_name* is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters **user_name** (*string*) – The name of the user to delete.

delete_user_policy (*user_name*, *policy_name*)

Deletes the specified policy document for the specified user.

Parameters

- **user_name** (*string*) – The name of the user the policy is associated with.
- **policy_name** (*string*) – The policy document to delete.

detach_group_policy (*policy_arn*, *group_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to detach
- **group_name** (*string*) – Group to detach the policy from

detach_role_policy (*policy_arn*, *role_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to detach
- **role_name** (*string*) – Role to detach the policy from

detach_user_policy (*policy_arn*, *user_name*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to detach
- **user_name** (*string*) – User to detach the policy from

enable_mfa_device (*user_name*, *serial_number*, *auth_code_1*, *auth_code_2*)

Enables the specified MFA device and associates it with the specified user.

Parameters

- **user_name** (*string*) – The username of the user
- **serial_number** (*string*) – The serial number which uniquely identifies the MFA device.
- **auth_code_1** (*string*) – An authentication code emitted by the device.
- **auth_code_2** (*string*) – A subsequent authentication code emitted by the device.

generate_credential_report ()

Generates a credential report for an account

A new credential report can only be generated every 4 hours. If one hasn't been generated in the last 4 hours then `get_credential_report` will error when called

get_account_alias ()

Get the alias for the current account.

This is referred to in the docs as `list_account_aliases`, but it seems you can only have one account alias currently.

For more information on account id aliases, please see <http://goo.gl/ToB7G>

get_account_password_policy ()

Returns the password policy for the AWS account.

get_account_summary ()

Get the alias for the current account.

This is referred to in the docs as `list_account_aliases`, but it seems you can only have one account alias currently.

For more information on account id aliases, please see <http://goo.gl/ToB7G>

get_all_access_keys (user_name, marker=None, max_items=None)

Get all access keys associated with an account.

Parameters

- **user_name** (*string*) – The username of the user
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you've received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_group_policies (group_name, marker=None, max_items=None)

List the names of the policies associated with the specified group.

Parameters

- **group_name** (*string*) – The name of the group the policy is associated with.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you've received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_groups (path_prefix='/', marker=None, max_items=None)

List the groups that have the specified path prefix.

Parameters

- **path_prefix** (*string*) – If provided, only groups whose paths match the provided prefix will be returned.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you've received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_mfa_devices (user_name, marker=None, max_items=None)

Get all MFA devices associated with an account.

Parameters

- **user_name** (*string*) – The username of the user

- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_server_certs (*path_prefix='/', marker=None, max_items=None*)

Lists the server certificates that have the specified path prefix. If none exist, the action returns an empty list.

Parameters

- **path_prefix** (*string*) – If provided, only certificates whose paths match the provided prefix will be returned.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_signing_certs (*marker=None, max_items=None, user_name=None*)

Get all signing certificates associated with an account.

If the *user_name* is not specified, it is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.
- **user_name** (*string*) – The username of the user

get_all_user_policies (*user_name, marker=None, max_items=None*)

List the names of the policies associated with the specified user.

Parameters

- **user_name** (*string*) – The name of the user the policy is associated with.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_all_users (*path_prefix='/', marker=None, max_items=None*)

List the users that have the specified path prefix.

Parameters

- **path_prefix** (*string*) – If provided, only users whose paths match the provided prefix will be returned.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.

- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_credential_report ()

Retrieves a credential report for an account

A report must have been generated in the last 4 hours to succeed. The report is returned as a base64 encoded blob within the response.

get_group (*group_name*, *marker=None*, *max_items=None*)

Return a list of users that are in the specified group.

Parameters

- **group_name** (*string*) – The name of the group whose information should be returned.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you've received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_group_policy (*group_name*, *policy_name*)

Retrieves the specified policy document for the specified group.

Parameters

- **group_name** (*string*) – The name of the group the policy is associated with.
- **policy_name** (*string*) – The policy document to get.

get_groups_for_user (*user_name*, *marker=None*, *max_items=None*)

List the groups that a specified user belongs to.

Parameters

- **user_name** (*string*) – The name of the user to list groups for.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you've received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

get_instance_profile (*instance_profile_name*)

Retrieves information about the specified instance profile, including the instance profile's path, GUID, ARN, and role.

Parameters **instance_profile_name** (*string*) – Name of the instance profile to get information about.

get_login_profiles (*user_name*)

Retrieves the login profile for the specified user.

Parameters **user_name** (*string*) – The username of the user

get_policy (*policy_arn*)

Get policy information.

Parameters **policy_arn** (*string*) – The ARN of the policy to get information for

get_policy_version (*policy_arn*, *version_id*)

Get policy information.

Parameters

- **policy_arn** (*string*) – The ARN of the policy to get information for a specific version
- **version_id** (*string*) – The id of the version to get information for

get_response (*action, params, path='/', parent=None, verb='POST', list_marker='Set'*)
Utility method to handle calls to IAM and parsing of responses.

get_role (*role_name*)
Retrieves information about the specified role, including the role's path, GUID, ARN, and the policy granting permission to EC2 to assume the role.
Parameters **role_name** (*string*) – Name of the role associated with the policy.

get_role_policy (*role_name, policy_name*)
Retrieves the specified policy document for the specified role.
Parameters

- **role_name** (*string*) – Name of the role associated with the policy.
- **policy_name** (*string*) – Name of the policy to get.

get_saml_provider (*saml_provider_arn*)
Returns the SAML provider metadocument that was uploaded when the provider was created or updated. This operation requires **'Signature Version 4'**.
Parameters **saml_provider_arn** (*string*) – The Amazon Resource Name (ARN) of the SAML provider to get information about.

get_server_certificate (*cert_name*)
Retrieves information about the specified server certificate.
Parameters **cert_name** (*string*) – The name of the server certificate you want to retrieve information about.

get_signin_url (*service='ec2'*)
Get the URL where IAM users can use their login profile to sign in to this account's console.
Parameters **service** (*string*) – Default service to go to in the console.

get_user (*user_name=None*)
Retrieve information about the specified user.

If the *user_name* is not specified, the *user_name* is determined implicitly based on the AWS Access Key ID used to sign the request.
Parameters **user_name** (*string*) – The name of the user to retrieve. If not specified, defaults to user making request.

get_user_policy (*user_name, policy_name*)
Retrieves the specified policy document for the specified user.
Parameters

- **user_name** (*string*) – The name of the user the policy is associated with.
- **policy_name** (*string*) – The policy document to get.

list_entities_for_policy (*policy_arn, path_prefix=None, marker=None, max_items=None, entity_filter=None*)

Parameters

- **policy_arn** (*string*) – The ARN of the policy to get entities for
- **marker** (*string*) – A marker used for pagination (received from previous accesses)
- **max_items** (*int*) – Send only max_items; allows paginations
- **path_prefix** (*string*) – Send only items prefixed by this path

- **entity_filter** (*string*) – Which entity type of User | Role | Group | LocalManagedPolicy | AWSManagedPolicy to return

list_instance_profiles (*path_prefix=None, marker=None, max_items=None*)

Lists the instance profiles that have the specified path prefix. If there are none, the action returns an empty list.

Parameters

- **path_prefix** (*string*) – The path prefix for filtering the results. For example: `/application_abc/component_xyz/`, which would get all instance profiles whose path starts with `/application_abc/component_xyz/`.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you've received a response where the results are truncated. Set it to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this parameter only when paginating results to indicate the maximum number of user names you want in the response.

list_instance_profiles_for_role (*role_name, marker=None, max_items=None*)

Lists the instance profiles that have the specified associated role. If there are none, the action returns an empty list.

Parameters

- **role_name** (*string*) – The name of the role to list instance profiles for.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you've received a response where the results are truncated. Set it to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this parameter only when paginating results to indicate the maximum number of user names you want in the response.

list_policies (*marker=None, max_items=None, only_attached=None, path_prefix=None, scope=None*)

List policies of account.

Parameters

- **marker** (*string*) – A marker used for pagination (received from previous accesses)
- **max_items** (*int*) – Send only max_items; allows paginations
- **only_attached** (*bool*) – Send only policies attached to other resources
- **path_prefix** (*string*) – Send only items prefixed by this path
- **scope** (*string*) – AWS|Local. Choose between AWS policies or your own

list_policy_versions (*policy_arn, marker=None, max_items=None*)

List policy versions.

Parameters

- **policy_arn** (*string*) – The ARN of the policy to get versions of
- **marker** (*string*) – A marker used for pagination (received from previous accesses)
- **max_items** (*int*) – Send only max_items; allows paginations

list_role_policies (*role_name*, *marker=None*, *max_items=None*)

Lists the names of the policies associated with the specified role. If there are none, the action returns an empty list.

Parameters

- **role_name** (*string*) – The name of the role to list policies for.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the marker element in the response you just received.
- **max_items** (*int*) – Use this parameter only when paginating results to indicate the maximum number of user names you want in the response.

list_roles (*path_prefix=None*, *marker=None*, *max_items=None*)

Lists the roles that have the specified path prefix. If there are none, the action returns an empty list.

Parameters

- **path_prefix** (*string*) – The path prefix for filtering the results.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the marker element in the response you just received.
- **max_items** (*int*) – Use this parameter only when paginating results to indicate the maximum number of user names you want in the response.

list_saml_providers ()

Lists the SAML providers in the account. This operation requires **‘Signature Version 4’_**.

list_server_certs (*path_prefix=''*, *marker=None*, *max_items=None*)

Lists the server certificates that have the specified path prefix. If none exist, the action returns an empty list.

Parameters

- **path_prefix** (*string*) – If provided, only certificates whose paths match the provided prefix will be returned.
- **marker** (*string*) – Use this only when paginating results and only in follow-up request after you’ve received a response where the results are truncated. Set this to the value of the Marker element in the response you just received.
- **max_items** (*int*) – Use this only when paginating results to indicate the maximum number of groups you want in the response.

put_group_policy (*group_name*, *policy_name*, *policy_json*)

Adds or updates the specified policy document for the specified group.

Parameters

- **group_name** (*string*) – The name of the group the policy is associated with.
- **policy_name** (*string*) – The policy document to get.
- **policy_json** (*string*) – The policy document.

put_role_policy (*role_name*, *policy_name*, *policy_document*)

Adds (or updates) a policy document associated with the specified role.

Parameters

- **role_name** (*string*) – Name of the role to associate the policy with.

- **policy_name** (*string*) – Name of the policy document.
- **policy_document** (*string*) – The policy document.

put_user_policy (*user_name, policy_name, policy_json*)

Adds or updates the specified policy document for the specified user.

Parameters

- **user_name** (*string*) – The name of the user the policy is associated with.
- **policy_name** (*string*) – The policy document to get.
- **policy_json** (*string*) – The policy document.

remove_role_from_instance_profile (*instance_profile_name, role_name*)

Removes the specified role from the specified instance profile.

Parameters

- **instance_profile_name** (*string*) – Name of the instance profile to update.
- **role_name** (*string*) – Name of the role to remove.

remove_user_from_group (*group_name, user_name*)

Remove a user from a group.

Parameters

- **group_name** (*string*) – The name of the group
- **user_name** (*string*) – The user to remove from the group.

resync_mfa_device (*user_name, serial_number, auth_code_1, auth_code_2*)

Synchronizes the specified MFA device with the AWS servers.

Parameters

- **user_name** (*string*) – The username of the user
- **serial_number** (*string*) – The serial number which uniquely identifies the MFA device.
- **auth_code_1** (*string*) – An authentication code emitted by the device.
- **auth_code_2** (*string*) – A subsequent authentication code emitted by the device.

set_default_policy_version (*policy_arn, version_id*)

Set default policy version.

Parameters

- **policy_arn** (*string*) – The ARN of the policy to set the default version for
- **version_id** (*string*) – The id of the version to set as default

update_access_key (*access_key_id, status, user_name=None*)

Changes the status of the specified access key from Active to Inactive or vice versa. This action can be used to disable a user's key as part of a key rotation workflow.

If the *user_name* is not specified, the *user_name* is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- **access_key_id** (*string*) – The ID of the access key.
- **status** (*string*) – Either Active or Inactive.
- **user_name** (*string*) – The username of user (optional).

update_account_password_policy (*allow_users_to_change_password=None*,
hard_expiry=None, *max_password_age=None*,
minimum_password_length=None, *password_reuse_prevention=None*,
require_lowercase_characters=None, *require_numbers=None*, *require_symbols=None*,
require_uppercase_characters=None)

Update the password policy for the AWS account.

Notes: unset parameters will be reset to Amazon default settings! Most of the password policy settings are enforced the next time your users change their passwords. When you set minimum length and character type requirements, they are enforced the next time your users change their passwords - users are not forced to change their existing passwords, even if the pre-existing passwords do not adhere to the updated password policy. When you set a password expiration period, the expiration period is enforced immediately.

Parameters

- **allow_users_to_change_password** (*bool*) – Allows all IAM users in your account to use the AWS Management Console to change their own passwords.
- **hard_expiry** (*bool*) – Prevents IAM users from setting a new password after their password has expired.
- **max_password_age** (*int*) – The number of days that an IAM user password is valid.
- **minimum_password_length** (*int*) – The minimum number of characters allowed in an IAM user password.
- **password_reuse_prevention** (*int*) – Specifies the number of previous passwords that IAM users are prevented from reusing.
- **require_lowercase_characters** (*bool*) – Specifies whether IAM user passwords must contain at least one lowercase character from the ISO basic Latin alphabet (a to z).
- **require_numbers** (*bool*) – Specifies whether IAM user passwords must contain at least one numeric character (0 to 9).
- **require_symbols** (*bool*) – Specifies whether IAM user passwords must contain at least one of the following non-alphanumeric characters: ! @ # \$ % ^ & * () _ + - = [] { } | ' .
- **require_uppercase_characters** (*bool*) – Specifies whether IAM user passwords must contain at least one uppercase character from the ISO basic Latin alphabet (A to Z).

update_assume_role_policy (*role_name*, *policy_document*)

Updates the policy that grants an entity permission to assume a role. Currently, only an Amazon EC2 instance can assume a role.

Parameters

- **role_name** (*string*) – Name of the role to update.
- **policy_document** (*string*) – The policy that grants an entity permission to assume the role.

update_group (*group_name*, *new_group_name=None*, *new_path=None*)

Updates name and/or path of the specified group.

Parameters

- **group_name** (*string*) – The name of the new group
- **new_group_name** (*string*) – If provided, the name of the group will be changed to this name.
- **new_path** (*string*) – If provided, the path of the group will be changed to this path.

update_login_profile (*user_name, password*)

Resets the password associated with the user’s login profile.

Parameters

- **user_name** (*string*) – The name of the user
- **password** (*string*) – The new password for the user

update_saml_provider (*saml_provider_arn, saml_metadata_document*)

Updates the metadata document for an existing SAML provider. This operation requires ‘Signature Version 4’.

Parameters

- **saml_provider_arn** (*string*) – The Amazon Resource Name (ARN) of the SAML provider to update.
- **saml_metadata_document** (*string*) – An XML document generated by an identity provider (IdP) that supports SAML 2.0. The document includes the issuer’s name, expiration information, and keys that can be used to validate the SAML authentication response (assertions) that are received from the IdP. You must generate the metadata document using the identity management software that is used as your organization’s IdP.

update_server_cert (*cert_name, new_cert_name=None, new_path=None*)

Updates the name and/or the path of the specified server certificate.

Parameters

- **cert_name** (*string*) – The name of the server certificate that you want to update.
- **new_cert_name** (*string*) – The new name for the server certificate. Include this only if you are updating the server certificate’s name.
- **new_path** (*string*) – If provided, the path of the certificate will be changed to this path.

update_signing_cert (*cert_id, status, user_name=None*)

Change the status of the specified signing certificate from Active to Inactive or vice versa.

If the *user_name* is not specified, it is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- **cert_id** (*string*) – The ID of the signing certificate
- **status** (*string*) – Either Active or Inactive.
- **user_name** (*string*) – The username of the user

update_user (*user_name, new_user_name=None, new_path=None*)

Updates name and/or path of the specified user.

Parameters

- **user_name** (*string*) – The name of the user

- **new_user_name** (*string*) – If provided, the username of the user will be changed to this username.
- **new_path** (*string*) – If provided, the path of the user will be changed to this path.

upload_server_cert (*cert_name, cert_body, private_key, cert_chain=None, path=None*)

Uploads a server certificate entity for the AWS Account. The server certificate entity includes a public key certificate, a private key, and an optional certificate chain, which should all be PEM-encoded.

Parameters

- **cert_name** (*string*) – The name for the server certificate. Do not include the path in this value.
- **cert_body** (*string*) – The contents of the public key certificate in PEM-encoded format.
- **private_key** (*string*) – The contents of the private key in PEM-encoded format.
- **cert_chain** (*string*) – The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.
- **path** (*string*) – The path for the server certificate.

upload_signing_cert (*cert_body, user_name=None*)

Uploads an X.509 signing certificate and associates it with the specified user.

If the *user_name* is not specified, it is determined implicitly based on the AWS Access Key ID used to sign the request.

Parameters

- **cert_body** (*string*) – The body of the signing certificate.
- **user_name** (*string*) – The username of the user

4.137.3 boto.iam.summarymap

```
class boto.iam.summarymap.SummaryMap (parent=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

4.138 API Reference

4.138.1 Kinesis

boto.kinesis

```
boto.kinesis.connect_to_region (region_name, **kw_params)
```

```
boto.kinesis.regions ()
```

Get all available regions for the Amazon Kinesis service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

boto.kinesis.layer1

class boto.kinesis.layer1.**KinesisConnection** (**kwargs)

Amazon Kinesis Service API Reference Amazon Kinesis is a managed service that scales elastically for real time processing of streaming big data.

APIVersion = '2013-12-02'

DefaultRegionEndpoint = 'kinesis.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.JSONResponseError*

ServiceName = 'Kinesis'

TargetPrefix = 'Kinesis_20131202'

add_tags_to_stream (*stream_name*, *tags*)

Adds or updates tags for the specified Amazon Kinesis stream. Each stream can have up to 10 tags.

If tags have already been assigned to the stream, *AddTagsToStream* overwrites any existing tags that correspond to the specified tag keys.

Parameters

- **stream_name** (*string*) – The name of the stream.
- **tags** (*map*) – The set of key-value pairs to use to create the tags.

create_stream (*stream_name*, *shard_count*)

Creates a Amazon Kinesis stream. A stream captures and transports data records that are continuously emitted from different data sources or producers. Scale-out within an Amazon Kinesis stream is explicitly supported by means of shards, which are uniquely identified groups of data records in an Amazon Kinesis stream.

You specify and control the number of shards that a stream is composed of. Each open shard can support up to 5 read transactions per second, up to a maximum total of 2 MB of data read per second. Each shard can support up to 1000 records written per second, up to a maximum total of 1 MB data written per second. You can add shards to a stream if the amount of data input increases and you can remove shards if the amount of data input decreases.

The stream name identifies the stream. The name is scoped to the AWS account used by the application. It is also scoped by region. That is, two streams in two different accounts can have the same name, and two streams in the same account, but in two different regions, can have the same name.

CreateStream is an asynchronous operation. Upon receiving a *CreateStream* request, Amazon Kinesis immediately returns and sets the stream status to *CREATING*. After the stream is created, Amazon Kinesis sets the stream status to *ACTIVE*. You should perform read and write operations only on an *ACTIVE* stream.

You receive a *LimitExceededException* when making a *CreateStream* request if you try to do one of the following:

- Have more than five streams in the *CREATING* state at any point in time.
- Create more shards than are authorized for your account.

The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, **'contact AWS Support'** to increase the limit on your account.

You can use *DescribeStream* to check the stream status, which is returned in *StreamStatus*.

CreateStream has a limit of 5 transactions per second per account.

Parameters

- **stream_name** (*string*) – A name to identify the stream. The stream name is scoped to the AWS account used by the application that creates the stream. It is also scoped by region. That is, two streams in two different AWS accounts can have the same name, and two streams in the same AWS account, but in two different regions, can have the same name.
- **shard_count** (*integer*) – The number of shards that the stream will use. The throughput of the stream is a function of the number of shards; more shards are required for greater provisioned throughput.

Note: The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, **‘contact AWS Support’** to increase the limit on your account.

delete_stream (*stream_name*)

Deletes a stream and all its shards and data. You must shut down any applications that are operating on the stream before you delete the stream. If an application attempts to operate on a deleted stream, it will receive the exception *ResourceNotFoundException*.

If the stream is in the *ACTIVE* state, you can delete it. After a *DeleteStream* request, the specified stream is in the *DELETING* state until Amazon Kinesis completes the deletion.

Note: Amazon Kinesis might continue to accept data read and write operations, such as *PutRecord*, *PutRecords*, and *GetRecords*, on a stream in the *DELETING* state until the stream deletion is complete.

When you delete a stream, any shards in that stream are also deleted, and any tags are dissociated from the stream.

You can use the *DescribeStream* operation to check the state of the stream, which is returned in *Stream-Status*.

DeleteStream has a limit of 5 transactions per second per account.

Parameters **stream_name** (*string*) – The name of the stream to delete.

describe_stream (*stream_name*, *limit=None*, *exclusive_start_shard_id=None*)

Describes the specified stream.

The information about the stream includes its current status, its Amazon Resource Name (ARN), and an array of shard objects. For each shard object, there is information about the hash key and sequence number ranges that the shard spans, and the IDs of any earlier shards that played in a role in creating the shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned when a record is put into the stream.

You can limit the number of returned shards using the *Limit* parameter. The number of shards in a stream may be too large to return from a single call to *DescribeStream*. You can detect this by using the *HasMoreShards* flag in the returned output. *HasMoreShards* is set to *True* when there is more data available.

DescribeStream is a paginated operation. If there are more shards available, you can request them using the shard ID of the last shard returned. Specify this ID in the *ExclusiveStartShardId* parameter in a subsequent request to *DescribeStream*.

DescribeStream has a limit of 10 transactions per second per account.

Parameters

- **stream_name** (*string*) – The name of the stream to describe.
- **limit** (*integer*) – The maximum number of shards to return.
- **exclusive_start_shard_id** (*string*) – The shard ID of the shard to start with.

get_records (*shard_iterator*, *limit=None*, *b64_decode=True*)

Gets data records from a shard.

Specify a shard iterator using the *ShardIterator* parameter. The shard iterator specifies the position in the shard from which you want to start reading data records sequentially. If there are no records available in the portion of the shard that the iterator points to, *GetRecords* returns an empty list. Note that it might take multiple calls to get to a portion of the shard that contains records.

You can scale by provisioning multiple shards. Your application should have one thread per shard, each reading continuously from its stream. To read from a stream continually, call *GetRecords* in a loop. Use *GetShardIterator* to get the shard iterator to specify in the first *GetRecords* call. *GetRecords* returns a new shard iterator in *NextShardIterator*. Specify the shard iterator returned in *NextShardIterator* in subsequent calls to *GetRecords*. Note that if the shard has been closed, the shard iterator can't return more data and *GetRecords* returns *null* in *NextShardIterator*. You can terminate the loop when the shard is closed, or when the shard iterator reaches the record with the sequence number or other attribute that marks it as the last record to process.

Each data record can be up to 50 KB in size, and each shard can read up to 2 MB per second. You can ensure that your calls don't exceed the maximum supported size or throughput by using the *Limit* parameter to specify the maximum number of records that *GetRecords* can return. Consider your average record size when determining this limit. For example, if your average record size is 40 KB, you can limit the data returned to about 1 MB per call by specifying 25 as the limit.

The size of the data returned by *GetRecords* will vary depending on the utilization of the shard. The maximum size of data that *GetRecords* can return is 10 MB. If a call returns 10 MB of data, subsequent calls made within the next 5 seconds throw *ProvisionedThroughputExceededException*. If there is insufficient provisioned throughput on the shard, subsequent calls made within the next 1 second throw *ProvisionedThroughputExceededException*. Note that *GetRecords* won't return any data when it throws an exception. For this reason, we recommend that you wait one second between calls to *GetRecords*; however, it's possible that the application will get exceptions for longer than 1 second.

To detect whether the application is falling behind in processing, add a timestamp to your records and note how long it takes to process them. You can also monitor how much data is in a stream using the CloudWatch metrics for write operations (*PutRecord* and *PutRecords*). For more information, see '[Monitoring Amazon Kinesis with Amazon CloudWatch](#)' in the Amazon Kinesis Developer Guide .

Parameters

- **shard_iterator** (*string*) – The position in the shard from which you want to start sequentially reading data records. A shard iterator specifies this position using the sequence number of a data record in the shard.
- **limit** (*integer*) – The maximum number of records to return. Specify a value of up to 10,000. If you specify a value that is greater than 10,000, *GetRecords* throws *InvalidArgumentException*.
- **b64_decode** (*boolean*) – Decode the Base64-encoded `Data` field of records.

get_shard_iterator (*stream_name*, *shard_id*, *shard_iterator_type*, *starting_sequence_number=None*)

Gets a shard iterator. A shard iterator expires five minutes after it is returned to the requester.

A shard iterator specifies the position in the shard from which to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in a shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned when a record is put into the stream.

You must specify the shard iterator type. For example, you can set the *ShardIteratorType* parameter to read exactly from the position denoted by a specific sequence number by using the *AT_SEQUENCE_NUMBER* shard iterator type, or right after the sequence number by using the *AFTER_SEQUENCE_NUMBER* shard iterator type, using sequence numbers returned by earlier calls to *PutRecord*, *PutRecords*, *GetRecords*,

or `DescribeStream`. You can specify the shard iterator type `TRIM_HORIZON` in the request to cause `ShardIterator` to point to the last untrimmed record in the shard in the system, which is the oldest data record in the shard. Or you can point to just after the most recent record in the shard, by using the shard iterator type `LATEST`, so that you always read the most recent data in the shard.

When you repeatedly read from an Amazon Kinesis stream use a `GetShardIterator` request to get the first shard iterator to use in your first `GetRecords` request and then use the shard iterator returned by the `GetRecords` request in `NextShardIterator` for subsequent reads. A new shard iterator is returned by every `GetRecords` request in `NextShardIterator`, which you use in the `ShardIterator` parameter of the next `GetRecords` request.

If a `GetShardIterator` request is made too often, you receive a `ProvisionedThroughputExceededException`. For more information about throughput limits, see `GetRecords`.

If the shard is closed, the iterator can't return more data, and `GetShardIterator` returns `null` for its `ShardIterator`. A shard can be closed using `SplitShard` or `MergeShards`.

`GetShardIterator` has a limit of 5 transactions per second per account per open shard.

Parameters

- `stream_name` (*string*) – The name of the stream.
- `shard_id` (*string*) – The shard ID of the shard to get the iterator for.
- `shard_iterator_type` (*string*) –

Determines how the shard iterator is used to start reading data records from the shard.

The following are the valid shard iterator types:

- **AT_SEQUENCE_NUMBER** - Start reading exactly from the position denoted by a specific sequence number.
- **AFTER_SEQUENCE_NUMBER** - Start reading right after the position denoted by a specific sequence number.
- **TRIM_HORIZON** - Start reading at the last untrimmed record in the shard in the system, which is the oldest data record in the shard.
- **LATEST** - Start reading just after the most recent record in the shard, so that you always read the most recent data in the shard.

Parameters `starting_sequence_number` (*string*) – The sequence number of the data record in the shard from which to start reading from.

Returns

A dictionary containing:

1. a `ShardIterator` with the value being the shard-iterator object

list_streams (*limit=None, exclusive_start_stream_name=None*)

Lists your streams.

The number of streams may be too large to return from a single call to `ListStreams`. You can limit the number of returned streams using the `Limit` parameter. If you do not specify a value for the `Limit` parameter, Amazon Kinesis uses the default limit, which is currently 10.

You can detect if there are more streams available to list by using the `HasMoreStreams` flag from the returned output. If there are more streams available, you can request more streams by using the name of the last stream returned by the `ListStreams` request in the `ExclusiveStartStreamName` parameter in a subsequent request to `ListStreams`. The group of stream names returned by the subsequent request is then added to the list. You can continue this process until all the stream names have been collected in the list.

`ListStreams` has a limit of 5 transactions per second per account.

Parameters

- **limit** (*integer*) – The maximum number of streams to list.
- **exclusive_start_stream_name** (*string*) – The name of the stream to start the list with.

list_tags_for_stream (*stream_name*, *exclusive_start_tag_key=None*, *limit=None*)

Lists the tags for the specified Amazon Kinesis stream.

Parameters

- **stream_name** (*string*) – The name of the stream.
- **exclusive_start_tag_key** (*string*) – The key to use as the starting point for the list of tags. If this parameter is set, *ListTagsForStream* gets all tags that occur after *ExclusiveStartTagKey*.
- **limit** (*integer*) – The number of tags to return. If this number is less than the total number of tags associated with the stream, *HasMoreTags* is set to *True*. To list additional tags, set *ExclusiveStartTagKey* to the last key in the response.

make_request (*action*, *body*)

Makes a request to the server, with stock multiple-retry logic.

merge_shards (*stream_name*, *shard_to_merge*, *adjacent_shard_to_merge*)

Merges two adjacent shards in a stream and combines them into a single shard to reduce the stream's capacity to ingest and transport data. Two shards are considered adjacent if the union of the hash key ranges for the two shards form a contiguous set with no gaps. For example, if you have two shards, one with a hash key range of 276...381 and the other with a hash key range of 382...454, then you could merge these two shards into a single shard that would have a hash key range of 276...454. After the merge, the single child shard receives data for all hash key values covered by the two parent shards.

MergeShards is called when there is a need to reduce the overall capacity of a stream because of excess capacity that is not being used. You must specify the shard to be merged and the adjacent shard for a stream. For more information about merging shards, see '[Merge Two Shards](#)' in the Amazon Kinesis Developer Guide .

If the stream is in the *ACTIVE* state, you can call *MergeShards*. If a stream is in the *CREATING*, *UPDATING*, or *DELETING* state, *MergeShards* returns a *ResourceInUseException*. If the specified stream does not exist, *MergeShards* returns a *ResourceNotFoundException*.

You can use *DescribeStream* to check the state of the stream, which is returned in *StreamStatus*.

MergeShards is an asynchronous operation. Upon receiving a *MergeShards* request, Amazon Kinesis immediately returns a response and sets the *StreamStatus* to *UPDATING*. After the operation is completed, Amazon Kinesis sets the *StreamStatus* to *ACTIVE*. Read and write operations continue to work while the stream is in the *UPDATING* state.

You use *DescribeStream* to determine the shard IDs that are specified in the *MergeShards* request.

If you try to operate on too many streams in parallel using *CreateStream*, *DeleteStream*, *MergeShards* or *SplitShard*, you will receive a *LimitExceededException*.

MergeShards has limit of 5 transactions per second per account.

Parameters

- **stream_name** (*string*) – The name of the stream for the merge.
- **shard_to_merge** (*string*) – The shard ID of the shard to combine with the adjacent shard for the merge.
- **adjacent_shard_to_merge** (*string*) – The shard ID of the adjacent shard for the merge.

`put_record` (*stream_name*, *data*, *partition_key*, *explicit_hash_key=None*, *sequence_number_for_ordering=None*, *exclusive_minimum_sequence_number=None*, *b64_encode=True*)

This operation puts a data record into an Amazon Kinesis stream from a producer. This operation must be called to send data from the producer into the Amazon Kinesis stream for real-time ingestion and subsequent processing. The *PutRecord* operation requires the name of the stream that captures, stores, and transports the data; a partition key; and the data blob itself. The data blob could be a segment from a log file, geographic/location data, website clickstream data, or any other data type.

The partition key is used to distribute data across shards. Amazon Kinesis segregates the data records that belong to a data stream into multiple shards, using the partition key associated with each data record to determine which shard a given data record belongs to.

Partition keys are Unicode strings, with a maximum length limit of 256 bytes. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards using the hash key ranges of the shards. You can override hashing the partition key to determine the shard by explicitly specifying a hash value using the *ExplicitHashKey* parameter. For more information, see the [‘Amazon Kinesis Developer Guide’](#).

PutRecord returns the shard ID of where the data record was placed and the sequence number that was assigned to the data record.

Sequence numbers generally increase over time. To guarantee strictly increasing ordering, use the *SequenceNumberForOrdering* parameter. For more information, see the [‘Amazon Kinesis Developer Guide’](#).

If a *PutRecord* request cannot be processed because of insufficient provisioned throughput on the shard involved in the request, *PutRecord* throws *ProvisionedThroughputExceededException*.

Data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream.

Parameters

- **stream_name** (*string*) – The name of the stream to put the data record into.
- **data** (*blob*) – The data blob to put into the record, which is Base64-encoded when the blob is serialized. The maximum size of the data blob (the payload after Base64-decoding) is 50 kilobytes (KB) Set *b64_encode* to disable automatic Base64 encoding.
- **partition_key** (*string*) – Determines which shard in the stream the data record is assigned to. Partition keys are Unicode strings with a maximum length limit of 256 bytes. Amazon Kinesis uses the partition key as input to a hash function that maps the partition key and associated data to a specific shard. Specifically, an MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key will map to the same shard within the stream.
- **explicit_hash_key** (*string*) – The hash value used to explicitly determine the shard the data record is assigned to by overriding the partition key hash.
- **sequence_number_for_ordering** (*string*) – Guarantees strictly increasing sequence numbers, for puts from the same client and to the same partition key. Usage: set the *SequenceNumberForOrdering* of record *n* to the sequence number of record *n-1* (as returned in the *PutRecordResult* when putting record *n-1*). If this parameter is not set, records will be coarsely ordered based on arrival time.
- **b64_encode** (*boolean*) – Whether to Base64 encode *data*. Can be set to `False` if *data* is already encoded to prevent double encoding.

put_records (*records*, *stream_name*, *b64_encode=True*)

Puts (writes) multiple data records from a producer into an Amazon Kinesis stream in a single call (also referred to as a *PutRecords* request). Use this operation to send data from a data producer into the Amazon Kinesis stream for real-time ingestion and processing. Each shard can support up to 1000 records written per second, up to a maximum total of 1 MB data written per second.

You must specify the name of the stream that captures, stores, and transports the data; and an array of request *Records*, with each record in the array requiring a partition key and data blob.

The data blob can be any type of data; for example, a segment from a log file, geographic/location data, website clickstream data, and so on.

The partition key is used by Amazon Kinesis as input to a hash function that maps the partition key and associated data to a specific shard. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream. For more information, see **‘Partition Key’** in the Amazon Kinesis Developer Guide .

Each record in the *Records* array may include an optional parameter, *ExplicitHashKey*, which overrides the partition key to shard mapping. This parameter allows a data producer to determine explicitly the shard where the record is stored. For more information, see **‘Adding Multiple Records with PutRecords’** in the Amazon Kinesis Developer Guide .

The *PutRecords* response includes an array of response *Records*. Each record in the response array directly correlates with a record in the request array using natural ordering, from the top to the bottom of the request and response. The response *Records* array always includes the same number of records as the request array.

The response *Records* array includes both successfully and unsuccessfully processed records. Amazon Kinesis attempts to process all records in each *PutRecords* request. A single record failure does not stop the processing of subsequent records.

A successfully-processed record includes *ShardId* and *SequenceNumber* values. The *ShardId* parameter identifies the shard in the stream where the record is stored. The *SequenceNumber* parameter is an identifier assigned to the put record, unique to all records in the stream.

An unsuccessfully-processed record includes *ErrorCode* and *ErrorMessage* values. *ErrorCode* reflects the type of error and can be one of the following values: *ProvisionedThroughputExceededException* or *InternalFailure*. *ErrorMessage* provides more detailed information about the *ProvisionedThroughputExceededException* exception including the account ID, stream name, and shard ID of the record that was throttled.

Data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream.

Parameters

- **records** (*list*) – The records associated with the request.
- **stream_name** (*string*) – The stream name associated with the request.
- **b64_encode** (*boolean*) – Whether to Base64 encode *data*. Can be set to `False` if *data* is already encoded to prevent double encoding.

remove_tags_from_stream (*stream_name*, *tag_keys*)

Deletes tags from the specified Amazon Kinesis stream.

If you specify a tag that does not exist, it is ignored.

Parameters

- **stream_name** (*string*) – The name of the stream.

- **tag_keys** (*list*) – A list of tag keys. Each corresponding tag is removed from the stream.

split_shard (*stream_name, shard_to_split, new_starting_hash_key*)

Splits a shard into two new shards in the stream, to increase the stream’s capacity to ingest and transport data. *SplitShard* is called when there is a need to increase the overall capacity of stream because of an expected increase in the volume of data records being ingested.

You can also use *SplitShard* when a shard appears to be approaching its maximum utilization, for example, when the set of producers sending data into the specific shard are suddenly sending more than previously anticipated. You can also call *SplitShard* to increase stream capacity, so that more Amazon Kinesis applications can simultaneously read data from the stream for real-time processing.

You must specify the shard to be split and the new hash key, which is the position in the shard where the shard gets split in two. In many cases, the new hash key might simply be the average of the beginning and ending hash key, but it can be any hash key value in the range being mapped into the shard. For more information about splitting shards, see ‘**Split a Shard**’_ in the Amazon Kinesis Developer Guide .

You can use *DescribeStream* to determine the shard ID and hash key values for the *ShardToSplit* and *NewStartingHashKey* parameters that are specified in the *SplitShard* request.

SplitShard is an asynchronous operation. Upon receiving a *SplitShard* request, Amazon Kinesis immediately returns a response and sets the stream status to *UPDATING*. After the operation is completed, Amazon Kinesis sets the stream status to *ACTIVE*. Read and write operations continue to work while the stream is in the *UPDATING* state.

You can use *DescribeStream* to check the status of the stream, which is returned in *StreamStatus*. If the stream is in the *ACTIVE* state, you can call *SplitShard*. If a stream is in *CREATING* or *UPDATING* or *DELETING* states, *DescribeStream* returns a *ResourceInUseException*.

If the specified stream does not exist, *DescribeStream* returns a *ResourceNotFoundException*. If you try to create more shards than are authorized for your account, you receive a *LimitExceededException*.

The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, ‘**contact AWS Support**’_ to increase the limit on your account.

If you try to operate on too many streams in parallel using *CreateStream*, *DeleteStream*, *MergeShards* or *SplitShard*, you receive a *LimitExceededException*.

SplitShard has limit of 5 transactions per second per account.

Parameters

- **stream_name** (*string*) – The name of the stream for the shard split.
- **shard_to_split** (*string*) – The shard ID of the shard to split.
- **new_starting_hash_key** (*string*) – A hash key value for the starting hash key of one of the child shards created by the split. The hash key range for a given shard constitutes a set of ordered contiguous positive integers. The value for *NewStartingHashKey* must be in the range of hash keys being mapped into the shard. The *NewStartingHashKey* hash key value and all higher hash key values in hash key range are distributed to one of the child shards. All the lower hash key values in the range are distributed to the other child shard.

boto.kinesis.exceptions

exception boto.kinesis.exceptions.**ExpiredIteratorException** (*status, reason, body=None, *args*)

exception boto.kinesis.exceptions.**InvalidArgumentException** (*status, reason, body=None, *args*)

```

exception boto.kinesis.exceptions.LimitExceededException (status, reason,
                                                         body=None, *args)
exception boto.kinesis.exceptions.ProvisionedThroughputExceededException (status,
                                                                              rea-
                                                                              son,
                                                                              body=None,
                                                                              *args)
exception boto.kinesis.exceptions.ResourceInUseException (status, reason,
                                                             body=None, *args)
exception boto.kinesis.exceptions.ResourceNotFoundException (status, reason,
                                                                body=None, *args)
exception boto.kinesis.exceptions.SubscriptionRequiredException (status, reason,
                                                                      body=None,
                                                                      *args)

```

4.138.2 manage

boto.manage

boto.manage.cmdshell

The cmdshell module uses the paramiko package to create SSH connections to the servers that are represented by instance objects. The module has functions for running commands, managing files, and opening interactive shell sessions over those connections.

```
class boto.manage.cmdshell.FakeServer (instance, ssh_key_file)
```

This object has a subset of the variables that are normally in a `boto.manage.server.Server` object. You can use this `FakeServer` object to create a `boto.manage.SSHClient` object if you don't have a real `Server` object.

Variables

- ***instance*** – A boto Instance object.
- ***ssh_key_file*** – The path to the SSH key file.

```
class boto.manage.cmdshell.LocalClient (server, host_key_file=None, uname='root')
```

Variables

- ***server*** – A Server object or FakeServer object.
- ***host_key_file*** – The path to the user's .ssh key files.
- ***uname*** – The username for the SSH connection. Default = 'root'.

```
close ()
```

```
exists (path)
```

Check for the specified path, or check a file at the specified path.

Return type boolean

Returns If the path or the file exist, the function returns True.

```
get_file (src, dst)
```

Copy a file from one directory to another.

```
isdir (path)
```

Check the specified path to determine if it is a directory.

Return type boolean

Returns Returns True if the path is an existing directory.

listdir (*path*)

List all of the files and subdirectories at the specified path.

Return type *list*

Returns Return a list containing the names of the entries in the directory given by path.

put_file (*src, dst*)

Copy a file from one directory to another.

run ()

Open a subprocess and run a command on the local host.

Return type *tuple*

Returns This function returns a tuple that contains an integer status and a string with the combined stdout and stderr output.

shell ()

class boto.manage.cmdshell.SSHClient (*server, host_key_file='~/.ssh/known_hosts', un-*
ame='root', timeout=None, ssh_pwd=None)

This class creates a paramiko.SSHClient() object that represents a session with an SSH server. You can use the SSHClient object to send commands to the remote host and manipulate files on the remote host.

Variables

- **server** – A Server object or FakeServer object.
- **host_key_file** – The path to the user's .ssh key files.
- **uname** – The username for the SSH connection. Default = 'root'.
- **timeout** – The optional timeout variable for the TCP connection.
- **ssh_pwd** – An optional password to use for authentication or for unlocking the private key.

close ()

Close an SSH session and any open channels that are tied to it.

connect (*num_retries=5*)

Connect to an SSH server and authenticate with it.

Parameters **num_retries** (*int*) – The maximum number of connection attempts.

exists (*path*)

Check the remote host for the specified path, or a file at the specified path. This function returns 1 if the path or the file exist on the remote host, and returns 0 if the path or the file does not exist on the remote host.

Parameters **path** (*string*) – The path to the directory or file that you want to check.

Return type *integer*

Returns If the path or the file exist, the function returns 1. If the path or the file do not exist on the remote host, the function returns 0.

get_file (*src, dst*)

Open an SFTP session on the remote host, and copy a file from the remote host to the specified path on the local host.

Parameters

- **src** (*string*) – The path to the target file on the remote host.
- **dst** (*string*) – The path on your local host where you want to store the file.

isdir (*path*)

Check the specified path on the remote host to determine if it is a directory.

Parameters **path** (*string*) – The path to the directory that you want to check.

Return type integer

Returns If the path is a directory, the function returns 1. If the path is a file or an invalid path, the function returns 0.

listdir (*path*)

List all of the files and subdirectories at the specified path on the remote host.

Parameters **path** (*string*) – The base path from which to obtain the list.

Return type list

Returns A list of files and subdirectories at the specified path.

open (*filename, mode='r', bufsize=-1*)

Open an SFTP session to the remote host, and open a file on that host.

Parameters

- **filename** (*string*) – The path to the file on the remote host.
- **mode** (*string*) – The file interaction mode.
- **bufsize** (*integer*) – The file buffer size.

Return type paramiko.sftp_file.SFTPFile

Returns A paramiko proxy object for a file on the remote server.

open_sftp ()

Open an SFTP session on the SSH server.

Return type paramiko.sftp_client.SFTPClient

Returns An SFTP client object.

put_file (*src, dst*)

Open an SFTP session on the remote host, and copy a file from the local host to the specified path on the remote host.

Parameters

- **src** (*string*) – The path to the target file on your local host.
- **dst** (*string*) – The path on the remote host where you want to store the file.

run (*command*)

Run a command on the remote host.

Parameters **command** (*string*) – The command that you want to send to the remote host.

Return type tuple

Returns This function returns a tuple that contains an integer status, the stdout from the command, and the stderr from the command.

run_pty (*command*)

Request a pseudo-terminal from a server, and execute a command on that server.

Parameters **command** (*string*) – The command that you want to run on the remote host.

Return type paramiko.channel.Channel

Returns An open channel object.

shell ()

Start an interactive shell session with the remote host.

```
boto.manage.cmdshell.sshclient_from_instance(instance, ssh_key_file,
                                             host_key_file='~/.ssh/known_hosts',
                                             user_name='root', ssh_pwd=None)
```

Create and return an SSHClient object given an instance object.

Parameters

- **instance** (*class`boto.ec2.instance.Instance` object*) – The instance object.
- **ssh_key_file** (*string*) – A path to the private key file that is used to log into the instance.
- **host_key_file** (*string*) – A path to the known_hosts file used by the SSH client. Defaults to ~/.ssh/known_hosts
- **user_name** (*string*) – The username to use when logging into the instance. Defaults to root.
- **ssh_pwd** (*string*) – The passphrase, if any, associated with private key.

```
boto.manage.cmdshell.start(server)
```

Connect to the specified server.

Returns If the server is local, the function returns a *boto.manage.cmdshell.LocalClient* object. If the server is remote, the function returns a *boto.manage.cmdshell.SSHClient* object.

boto.manage.propget

```
boto.manage.propget.get(prop, choices=None)
```

boto.manage.server

boto.manage.task

boto.manage.volume

4.138.3 mturk

boto.mturk

boto.mturk.connection

```
class boto.mturk.connection.Assignment(connection)
```

Class to extract an Assignment structure from a response (used in ResultSet)

Will have attributes named as per the Developer Guide, e.g. AssignmentId, WorkerId, HITId, Answer, etc

```
endElement(name, value, connection)
```

```
class boto.mturk.connection.BaseAutoResultElement(connection)
```

Base class to automatically add attributes when parsing XML

```
endElement(name, value, connection)
```

```
startElement(name, attrs, connection)
```

```
class boto.mturk.connection.FileUploadURL(connection)
```

Class to extract an FileUploadURL structure from a response

class boto.mturk.connection.HIT (*connection*)

Class to extract a HIT structure from a response (used in ResultSet)

Will have attributes named as per the Developer Guide, e.g. HITId, HITTypeId, CreationTime

expired

Has this HIT expired yet?

class boto.mturk.connection.HITTypeId (*connection*)

Class to extract an HITTypeId structure from a response

class boto.mturk.connection.MTurkConnection (*aws_access_key_id=None,*
aws_secret_access_key=None,
is_secure=True, port=None, proxy=None,
proxy_port=None, proxy_user=None,
proxy_pass=None, host=None, debug=0,
https_connection_factory=None, secu-
rity_token=None, profile_name=None)

APIVersion = '2014-08-15'

approve_assignment (*assignment_id, feedback=None*)

approve_rejected_assignment (*assignment_id, feedback=None*)

assign_qualification (*qualification_type_id, worker_id, value=1, send_notification=True*)

block_worker (*worker_id, reason*)

Block a worker from working on my tasks.

change_hit_type_of_hit (*hit_id, hit_type*)

Change the HIT type of an existing HIT. Note that the reward associated with the new HIT type must match the reward of the current HIT type in order for the operation to be valid.

create_hit (*hit_type=None, question=None, hit_layout=None, lifetime=datetime.timedelta(7),*
max_assignments=1, title=None, description=None, keywords=None, reward=None,
duration=datetime.timedelta(7), approval_delay=None, annotation=None, ques-
tions=None, qualifications=None, layout_params=None, response_groups=None)

Creates a new HIT. Returns a ResultSet See: http://docs.amazonwebservices.com/AWSMechTurk/2012-03-25/AWSMturkAPI/ApiReference_CreateHITOperation.html

create_qualification_type (*name, description, status, keywords=None,*
retry_delay=None, test=None, answer_key=None, an-
swer_key_xml=None, test_duration=None, auto_granted=False,
auto_granted_value=1)

Create a new Qualification Type.

name: This will be visible to workers and must be unique for a given requester.

description: description shown to workers. Max 2000 characters.

status: 'Active' or 'Inactive'

keywords: list of keyword strings or comma separated string. Max length of 1000 characters when concatenated with commas.

retry_delay: number of seconds after requesting a qualification the worker must wait before they can ask again. If not specified, workers can only request this qualification once.

test: a QuestionForm

answer_key: an XML string of your answer key, for automatically scored qualification tests. (Consider implementing an AnswerKey class for this to support.)

test_duration: the number of seconds a worker has to complete the test.

auto_granted: if True, requests for the Qualification are granted immediately. Can't coexist with a test.

auto_granted_value: auto_granted qualifications are given this value.

disable_hit (*hit_id*, *response_groups=None*)

Remove a HIT from the Mechanical Turk marketplace, approves all submitted assignments that have not already been approved or rejected, and disposes of the HIT and all assignment data.

Assignments for the HIT that have already been submitted, but not yet approved or rejected, will be automatically approved. Assignments in progress at the time of the call to DisableHIT will be approved once the assignments are submitted. You will be charged for approval of these assignments. DisableHIT completely disposes of the HIT and all submitted assignment data. Assignment results data cannot be retrieved for a HIT that has been disposed.

It is not possible to re-enable a HIT once it has been disabled. To make the work from a disabled HIT available again, create a new HIT.

dispose_hit (*hit_id*)

Dispose of a HIT that is no longer needed.

Only HITs in the "reviewable" state, with all submitted assignments approved or rejected, can be disposed. A Requester can call GetReviewableHITs to determine which HITs are reviewable, then call GetAssignmentsForHIT to retrieve the assignments. Disposing of a HIT removes the HIT from the results of a call to GetReviewableHITs.

dispose_qualification_type (*qualification_type_id*)

TODO: Document.

static duration_as_seconds (*duration*)

expire_hit (*hit_id*)

Expire a HIT that is no longer needed.

The effect is identical to the HIT expiring on its own. The HIT no longer appears on the Mechanical Turk web site, and no new Workers are allowed to accept the HIT. Workers who have accepted the HIT prior to expiration are allowed to complete it or return it, or allow the assignment duration to elapse (abandon the HIT). Once all remaining assignments have been submitted, the expired HIT becomes "reviewable", and will be returned by a call to GetReviewableHITs.

extend_hit (*hit_id*, *assignments_increment=None*, *expiration_increment=None*)

Increase the maximum number of assignments, or extend the expiration date, of an existing HIT.

NOTE: If a HIT has a status of Reviewable and the HIT is extended to make it Available, the HIT will not be returned by GetReviewableHITs, and its submitted assignments will not be returned by GetAssignmentsForHIT, until the HIT is Reviewable again. Assignment auto-approval will still happen on its original schedule, even if the HIT has been extended. Be sure to retrieve and approve (or reject) submitted assignments before extending the HIT, if so desired.

get_account_balance ()

get_all_hits ()

Return all of a Requester's HITs

Despite what search_hits says, it does not return all hits, but instead returns a page of hits. This method will pull the hits from the server 100 at a time, but will yield the results iteratively, so subsequent requests are made on demand.

get_all_qualifications_for_qual_type (*qualification_type_id*)

get_assignment (*assignment_id*, *response_groups=None*)

Retrieves an assignment using the assignment's ID. Requesters can only retrieve their own assignments, and only assignments whose related HIT has not been disposed.

The returned ResultSet will have the following attributes:

Request This element is present only if the Request ResponseGroup is specified.

Assignment The assignment. The response includes one Assignment object.

HIT The HIT associated with this assignment. The response includes one HIT object.

get_assignments (*hit_id*, *status=None*, *sort_by='SubmitTime'*, *sort_direction='Ascending'*,
page_size=10, *page_number=1*, *response_groups=None*)

Retrieves completed assignments for a HIT. Use this operation to retrieve the results for a HIT.

The returned ResultSet will have the following attributes:

NumResults The number of assignments on the page in the filtered results list, equivalent to the number of assignments being returned by this call. A non-negative integer, as a string.

PageNumber The number of the page in the filtered results list being returned. A positive integer, as a string.

TotalNumResults The total number of HITs in the filtered results list based on this call. A non-negative integer, as a string.

The ResultSet will contain zero or more Assignment objects

get_file_upload_url (*assignment_id*, *question_idenfifier*)

Generates and returns a temporary URL to an uploaded file. The temporary URL is used to retrieve the file as an answer to a FileUploadAnswer question, it is valid for 60 seconds.

Will have a FileUploadURL attribute as per the API Reference.

get_help (*about*, *help_type='Operation'*)

Return information about the Mechanical Turk Service operations and response group NOTE - this is basically useless as it just returns the URL of the documentation

help_type: either 'Operation' or 'ResponseGroup'

get_hit (*hit_id*, *response_groups=None*)

static get_keywords_as_string (*keywords*)

Returns a comma+space-separated string of keywords from either a list or a string

static get_price_as_price (*reward*)

Returns a Price data structure from either a float or a Price

get_qualification_requests (*qualification_type_id*, *sort_by='Expiration'*,
sort_direction='Ascending', *page_size=10*, *page_number=1*)

TODO: Document.

get_qualification_score (*qualification_type_id*, *worker_id*)

TODO: Document.

get_qualification_type (*qualification_type_id*)

get_qualifications_for_qualification_type (*qualification_type_id*, *page_size=100*,
page_number=1)

get_reviewable_hits (*hit_type=None*, *status='Reviewable'*, *sort_by='Expiration'*,
sort_direction='Ascending', *page_size=10*, *page_number=1*)

Retrieve the HITs that have a status of Reviewable, or HITs that have a status of Reviewing, and that belong to the Requester calling the operation.

grant_bonus (*worker_id*, *assignment_id*, *bonus_price*, *reason*)

Issues a payment of money from your account to a Worker. To be eligible for a bonus, the Worker must have submitted results for one of your HITs, and have had those results approved or rejected. This payment happens separately from the reward you pay to the Worker when you approve the Worker's assignment. The Bonus must be passed in as an instance of the Price object.

grant_qualification (*qualification_request_id*, *integer_value=1*)

TODO: Document.

notify_workers (*worker_ids, subject, message_text*)

Send a text message to workers.

register_hit_type (*title, description, reward, duration, keywords=None, approval_delay=None, qual_req=None*)

Register a new HIT Type title, description are strings reward is a Price object duration can be a timedelta, or an object castable to an int

reject_assignment (*assignment_id, feedback=None*)

revoke_qualification (*subject_id, qualification_type_id, reason=None*)

TODO: Document.

search_hits (*sort_by='CreationTime', sort_direction='Ascending', page_size=10, page_number=1, response_groups=None*)

Return a page of a Requester's HITs, on behalf of the Requester. The operation returns HITs of any status, except for HITs that have been disposed with the DisposeHIT operation. Note: The SearchHITs operation does not accept any search parameters that filter the results.

search_qualification_types (*query=None, sort_by='Name', sort_direction='Ascending', page_size=10, page_number=1, must_be_requestable=True, must_be_owned_by_caller=True*)

TODO: Document.

send_test_event_notification (*hit_type, url, event_types=None, test_event_type='Ping'*)

Performs a SendTestEventNotification operation with REST notification for a specified HIT type

set_email_notification (*hit_type, email, event_types=None*)

Performs a SetHITTypeNotification operation to set email notification for a specified HIT type

set_rest_notification (*hit_type, url, event_types=None*)

Performs a SetHITTypeNotification operation to set REST notification for a specified HIT type

set_reviewing (*hit_id, revert=None*)

Update a HIT with a status of Reviewable to have a status of Reviewing, or reverts a Reviewing HIT back to the Reviewable status.

Only HITs with a status of Reviewable can be updated with a status of Reviewing. Similarly, only Reviewing HITs can be reverted back to a status of Reviewable.

set_sqs_notification (*hit_type, queue_url, event_types=None*)

Performs a SetHITTypeNotification operation so set SQS notification for a specified HIT type. Queue URL is of form: https://queue.amazonaws.com/<CUSTOMER_ID>/<QUEUE_NAME> and can be found when looking at the details for a Queue in the AWS Console

unblock_worker (*worker_id, reason*)

Unblock a worker from working on my tasks.

update_qualification_score (*qualification_type_id, worker_id, value*)

TODO: Document.

update_qualification_type (*qualification_type_id, description=None, status=None, retry_delay=None, test=None, answer_key=None, test_duration=None, auto_granted=None, auto_granted_value=None*)

exception `boto.mturk.connection.MTurkRequestError` (*status, reason, body=None*)

Error for MTurk Requests

class `boto.mturk.connection.Qualification` (*connection*)

Class to extract an Qualification structure from a response (used in ResultSet)

Will have attributes named as per the Developer Guide such as `QualificationTypeId`, `IntegerValue`. Does not seem to contain `GrantTime`.

```
class boto.mturk.connection.QualificationRequest (connection)
    Class to extract an QualificationRequest structure from a response (used in ResultSet)
```

Will have attributes named as per the Developer Guide, e.g. `QualificationRequestId`, `QualificationTypeId`, `SubjectId`, etc

```
    endElement (name, value, connection)
```

```
class boto.mturk.connection.QualificationType (connection)
    Class to extract an QualificationType structure from a response (used in ResultSet)
```

Will have attributes named as per the Developer Guide, e.g. `QualificationTypeId`, `CreationTime`, `Name`, etc

```
class boto.mturk.connection.QuestionFormAnswer (connection)
    Class to extract Answers from inside the embedded XML QuestionFormAnswers element inside the Answer element which is part of the Assignment and QualificationRequest structures
```

A `QuestionFormAnswers` element contains an `Answer` element for each question in the HIT or Qualification test for which the Worker provided an answer. Each `Answer` contains a `QuestionIdentifier` element whose value corresponds to the `QuestionIdentifier` of a `Question` in the `QuestionForm`. See the `QuestionForm` data structure for more information about questions and answer specifications.

If the question expects a free-text answer, the `Answer` element contains a `FreeText` element. This element contains the Worker's answer

NOTE - currently really only supports free-text and selection answers

```
    endElement (name, value, connection)
```

boto.mturk.layoutparam

```
class boto.mturk.layoutparam.LayoutParameter (name, value)
    Representation of a single HIT layout parameter
```

```
    get_as_params ()
```

```
class boto.mturk.layoutparam.LayoutParameters (layoutParameters=None)
```

```
    add (req)
```

```
    get_as_params ()
```

boto.mturk.notification

Provides `NotificationMessage` and `Event` classes, with utility methods, for implementations of the Mechanical Turk Notification API.

```
class boto.mturk.notification.Event (d)
```

```
class boto.mturk.notification.NotificationMessage (d)
    Constructor; expects parameter d to be a dict of string parameters from a REST transport notification message
```

```
    EVENT_PATTERN = 'Event\\.(?P<n>\\d+)\\. (?P<param>\\w+)'
```

```
    EVENT_RE = <_sre.SRE_Pattern object>
```

```
    NOTIFICATION_VERSION = '2006-05-05'
```

```
    NOTIFICATION_WSDL = 'http://mechanicalturk.amazonaws.com/AWSMechanicalTurk/2006-05-05/'
```

```
OPERATION_NAME = 'Notify'  
SERVICE_NAME = 'AWSMechanicalTurkRequesterNotification'
```

```
verify(secret_key)
```

Verifies the authenticity of a notification message.

TODO: This is doing a form of authentication and this functionality should really be merged with the pluggable authentication mechanism at some point.

boto.mturk.price

```
class boto.mturk.price.Price(amount=0.0, currency_code='USD')
```

```
endElement(name, value, connection)
```

```
get_as_params(label, ord=1)
```

```
startElement(name, attrs, connection)
```

boto.mturk.qualification

```
class boto.mturk.qualification.AdultRequirement(comparator, integer_value, re-  
quired_to_preview=False)
```

Requires workers to acknowledge that they are over 18 and that they agree to work on potentially offensive content. The value type is boolean, 1 (required), 0 (not required, the default).

```
class boto.mturk.qualification.LocaleRequirement(comparator, locale, re-  
quired_to_preview=False)
```

A Qualification requirement based on the Worker's location. The Worker's location is specified by the Worker to Mechanical Turk when the Worker creates his account.

```
get_as_params()
```

```
class boto.mturk.qualification.NumberHitsApprovedRequirement(comparator,  
integer_value, re-  
quired_to_preview=False)
```

Specifies the total number of HITs submitted by a Worker that have been approved. The value is an integer greater than or equal to 0.

If specifying a Country and Subdivision, use a tuple of valid ISO 3166 country code and ISO 3166-2 subdivision code, e.g. ('US', 'CA') for the US State of California.

When using the 'In' and 'NotIn', locale should be a list of Countries and/or (Country, Subdivision) tuples.

```
class boto.mturk.qualification.PercentAssignmentsAbandonedRequirement(comparator,  
inte-  
ger_value,  
re-  
quired_to_preview=False)
```

The percentage of assignments the Worker has abandoned (allowed the deadline to elapse), over all assignments the Worker has accepted. The value is an integer between 0 and 100.

```
class boto.mturk.qualification.PercentAssignmentsApprovedRequirement(comparator,  
inte-  
ger_value,  
re-  
quired_to_preview=False)
```

The percentage of assignments the Worker has submitted that were subsequently approved by the Requester, over all assignments the Worker has submitted. The value is an integer between 0 and 100.

```
class boto.mturk.qualification.PercentAssignmentsRejectedRequirement (comparator,
                                                                    inte-
                                                                    ger_value,
                                                                    re-
                                                                    quired_to_preview=False)
```

The percentage of assignments the Worker has submitted that were subsequently rejected by the Requester, over all assignments the Worker has submitted. The value is an integer between 0 and 100.

```
class boto.mturk.qualification.PercentAssignmentsReturnedRequirement (comparator,
                                                                    inte-
                                                                    ger_value,
                                                                    re-
                                                                    quired_to_preview=False)
```

The percentage of assignments the Worker has returned, over all assignments the Worker has accepted. The value is an integer between 0 and 100.

```
class boto.mturk.qualification.PercentAssignmentsSubmittedRequirement (comparator,
                                                                    inte-
                                                                    ger_value,
                                                                    re-
                                                                    quired_to_preview=False)
```

The percentage of assignments the Worker has submitted, over all assignments the Worker has accepted. The value is an integer between 0 and 100.

```
class boto.mturk.qualification.Qualifications (requirements=None)
```

```
    add (req)
```

```
    get_as_params ()
```

```
class boto.mturk.qualification.Requirement (qualification_type_id,          compara-
                                                                    tor,          integer_value=None,          re-
                                                                    quired_to_preview=False)
```

Representation of a single requirement

```
    get_as_params ()
```

boto.mturk.question

```
class boto.mturk.question.AnswerSpecification (spec)
```

```
    get_as_xml ()
```

```
    template = '<AnswerSpecification>% (spec) s</AnswerSpecification>'
```

```
class boto.mturk.question.Application (width, height, **parameters)
```

```
    get_as_xml ()
```

```
    get_inner_content (content)
```

```
    parameter_template = '<Name>% (name) s</Name><Value>% (value) s</Value>'
```

```
    template = '<Application><% (class_) s>% (content) s</% (class_) s></Application>'
```

```
class boto.mturk.question.Binary (type, subtype, url, alt_text)
```

```
    template = '<Binary><MimeType><Type>% (type) s</Type><SubType>% (subtype) s</SubType></Mime
```

```
class boto.mturk.question.Constraint

    get_as_xml()
    get_attributes()

class boto.mturk.question.Constraints

    get_as_xml()
    template = '<Constraints>% (content) s</Constraints>'

class boto.mturk.question.ExternalQuestion (external_url, frame_height)
    An object for constructing an External Question.

    get_as_params (label='ExternalQuestion')
    get_as_xml()
    schema_url = 'http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2006-07-11/ExternalQuestion.xsd'
    template = '<ExternalQuestion xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2006-07-11/ExternalQuestion.xsd">% (content) s</ExternalQuestion>'

class boto.mturk.question.FileUploadAnswer (min_bytes, max_bytes)

    get_as_xml()
    template = '<FileUploadAnswer><MaxFileSizeInBytes>% (max_bytes) d</MaxFileSizeInBytes><MaxFileCount>% (max_file_count) d</MaxFileCount><FileUploads>% (file_uploads) s</FileUploads></FileUploadAnswer>'

class boto.mturk.question.Flash (url, *args, **kwargs)

    get_inner_content (content)

class boto.mturk.question.FormattedContent (content)

    schema_url = 'http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2006-07-11/FormattedContent.xsd'
    template = '<FormattedContent><![CDATA[% (content) s]]></FormattedContent>'

class boto.mturk.question.FreeTextAnswer (default=None, constraints=None, num_lines=None)

    get_as_xml()
    template = '<FreeTextAnswer>% (items) s</FreeTextAnswer>'

class boto.mturk.question.HTMLQuestion (html_form, frame_height)

    get_as_params (label='HTMLQuestion')
    get_as_xml()
    schema_url = 'http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2011-11-01/HTMLQuestion.xsd'
    template = '<HTMLQuestion xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2011-11-01/HTMLQuestion.xsd">% (html_form) s</HTMLQuestion>'

class boto.mturk.question.JavaApplet (path, filename, *args, **kwargs)

    get_inner_content (content)
```

```

class boto.mturk.question.LengthConstraint (min_length=None, max_length=None)

    attribute_names = ('minLength', 'maxLength')
    template = '<Length %(attrs)s />'

class boto.mturk.question.List
    A bulleted list suitable for OrderedContent or Overview content

    get_as_xml()

class boto.mturk.question.NumberOfLinesSuggestion (num_lines=1)

    get_as_xml()
    template = '<NumberOfLinesSuggestion>%(num_lines)s</NumberOfLinesSuggestion>'

class boto.mturk.question.NumericConstraint (min_value=None, max_value=None)

    attribute_names = ('minValue', 'maxValue')
    template = '<IsNumeric %(attrs)s />'

class boto.mturk.question.OrderedContent

    append_field (field, value)
    get_as_xml()

class boto.mturk.question.Overview

    get_as_params (label='Overview')
    get_as_xml()
    template = '<Overview>%(content)s</Overview>'

class boto.mturk.question.Question (identifier, content, answer_spec, is_required=False, display_name=None)

    get_as_params (label='Question')
    get_as_xml()
    template = '<Question>%(items)s</Question>'

class boto.mturk.question.QuestionContent

    get_as_xml()
    template = '<QuestionContent>%(content)s</QuestionContent>'

class boto.mturk.question.QuestionForm
    From the AMT API docs:

    The top-most element of the QuestionForm data structure is a QuestionForm element. This element contains optional Overview elements and one or more Question elements. There can be any number of these two element types listed in any order. The following example structure has an Overview element and a Question element followed by a second Overview element and Question element—all within the same QuestionForm.

```

```

<QuestionForm xmlns="[the QuestionForm schema URL]">
  <Overview>
    [...]
  </Overview>
  <Question>
    [...]
  </Question>
  <Overview>
    [...]
  </Overview>
  <Question>
    [...]
  </Question>
  [...]
</QuestionForm>

```

QuestionForm is implemented as a list, so to construct a QuestionForm, simply append Questions and Overviews (with at least one Question).

```
get_as_xml()
```

```
is_valid()
```

```
schema_url = 'http://mechanicalturk.amazonaws.com/AWSMechanicalTurkDataSchemas/2005-10-
```

```
xml_template = '<QuestionForm xmlns="http://mechanicalturk.amazonaws.com/AWSMechanical'
```

```
class boto.mturk.question.RegExConstraint (pattern, error_text=None, flags=None)
```

```
attribute_names = ('regex', 'errorText', 'flags')
```

```
get_attributes()
```

```
template = '<AnswerFormatRegex %(attrs)s />'
```

```
class boto.mturk.question.SelectionAnswer (min=1, max=1, style=None, selections=None,
                                           type='text', other=False)
```

A class to generate SelectionAnswer XML data structures. Does not yet implement Binary selection options.

```
ACCEPTED_STYLES = ['radiobutton', 'dropdown', 'checkbox', 'list', 'combobox', 'multich'
```

```
MAX_SELECTION_COUNT_XML_TEMPLATE = '<MaxSelectionCount>%s</MaxSelectionCount>'
```

```
MIN_SELECTION_COUNT_XML_TEMPLATE = '<MinSelectionCount>%s</MinSelectionCount>'
```

```
OTHER_SELECTION_ELEMENT_NAME = 'OtherSelection'
```

```
SELECTIONANSWER_XML_TEMPLATE = '<SelectionAnswer>%s<Selections>%s</Selections></Sele'
```

```
SELECTION_VALUE_XML_TEMPLATE = '<%s>%s</%s>'
```

```
SELECTION_XML_TEMPLATE = '<Selection><SelectionIdentifier>%s</SelectionIdentifier>%s</'
```

```
STYLE_XML_TEMPLATE = '<StyleSuggestion>%s</StyleSuggestion>'
```

```
get_as_xml()
```

```
class boto.mturk.question.SimpleField (field, value)
```

A Simple name/value pair that can be easily rendered as XML.

```
>>> SimpleField('Text', 'A text string').get_as_xml()
'<Text>A text string</Text>'
```

```
template = '<%(field)s>%(value)s</%(field)s>'
```



```
class boto.mturk.question.ValidatingXML
```

```
    validate()
```

```
class boto.mturk.question.XMLTemplate
```

```
    get_as_xml()
```

4.138.4 mws

boto.mws

boto.mws.connection

```
class boto.mws.connection.MWSConnection (*args, **kw)
```

```
    class ResponseErrorFactory (scopes=None)
```

```
    class ResponseFactory (scopes=None)
```

```
        element_factory (name, parent)
```

```
        find_element (action, suffix, parent)
```

```
        search_scopes (key)
```

```
    authorize (**kw)
```

MWS Authorize/2013-01-01 API call; quota=10 restore=1.00 Reserves a specified amount against the payment method(s) stored in the order reference.

ElementIterMap: AuthorizationAmount (ResponseElement or anything iterable/dict-like) Required: AmazonOrderReferenceId+AuthorizationReferenceId+AuthorizationAmount

```
    cancel_feed_submissions (*args, **kw)
```

MWS CancelFeedSubmissions/2009-01-01 API call; quota=10 restore=45.00 Cancels one or more feed submissions and returns a

count of the feed submissions that were canceled.

Lists: FeedSubmissionIdList.Id, FeedTypeList.Type

```
    cancel_fulfillment_order (**kw)
```

MWS CancelFulfillmentOrder/2010-10-01 API call; quota=30 restore=0.50 Requests that Amazon stop attempting to fulfill an existing fulfillment order.

Required: SellerFulfillmentOrderId

```
    cancel_order_reference (**kw)
```

MWS CancelOrderReference/2013-01-01 API call; quota=10 restore=1.00 Cancel an order reference; all authorizations associated with

this order reference are also closed.

Required: AmazonOrderReferenceId

```
    cancel_report_requests (*args, **kw)
```

MWS CancelReportRequests/2009-01-01 API call; quota=10 restore=45.00 Cancel one or more report requests, returning the count of the

canceled report requests and the report request information.

capture (**kw)

MWS Capture/2013-01-01 API call; quota=10 restore=1.00 Captures funds from an authorized payment instrument.

ElementIterMap: CaptureAmount (ResponseElement or anything iterable/dict-like) Required: AmazonAuthorizationId+CaptureReferenceId+CaptureAmount

close_authorization (**kw)

MWS CloseAuthorization/2013-01-01 API call; quota=10 restore=1.00 Closes an authorization.

Required: AmazonAuthorizationId

close_order_reference (**kw)

MWS CloseOrderReference/2013-01-01 API call; quota=10 restore=1.00 Confirms that an order reference has been fulfilled (fully or partially) and that you do not expect to create any new authorizations on this order reference.

Required: AmazonOrderReferenceId

confirm_order_reference (**kw)

MWS ConfirmOrderReference/2013-01-01 API call; quota=10 restore=1.00 Confirms that the order reference is free of constraints and all required information has been set on the order reference.

Required: AmazonOrderReferenceId

create_fulfillment_order (**kw)

MWS CreateFulfillmentOrder/2010-10-01 API call; quota=30 restore=0.50 Requests that Amazon ship items from the seller's inventory to a destination address.

ElementIterMap: DestinationAddress, Items (ResponseElement or anything iterable/dict-like) Required: SellerFulfillmentOrderId+DisplayableOrderId+ShippingSpeedCategory+DisplayableOrderDateTime+DestinationAddress+

create_inbound_shipment (**kw)

MWS CreateInboundShipment/2010-10-01 API call; quota=30 restore=0.50 Creates an inbound shipment.

ElementIterMap: InboundShipmentHeader, InboundShipmentItems (ResponseElement or anything iterable/dict-like) Required: ShipmentId+InboundShipmentHeader+InboundShipmentItems

create_inbound_shipment_plan (**kw)

MWS CreateInboundShipmentPlan/2010-10-01 API call; quota=30 restore=0.50 Returns the information required to create an inbound shipment.

ElementIterMap: ShipFromAddress, InboundShipmentPlanRequestItems (ResponseElement or anything iterable/dict-like) Required: ShipFromAddress+InboundShipmentPlanRequestItems

create_subscription (**kw)

MWS CreateSubscription/2013-07-01 API call; quota=25 restore=0.50 Creates a new subscription for the specified notification type and destination.

ElementIterMap: Subscription (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+Subscription

delete_subscription (**kw)

MWS DeleteSubscription/2013-07-01 API call; quota=25 restore=0.50 Deletes the subscription for the specified notification type and destination.

ElementIterMap: Destination (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+NotificationType+Destination

deregister_destination (**kw)

MWS DeregisterDestination/2013-07-01 API call; quota=25 restore=0.50 Removes an existing destination from the list of registered destinations.

ElementIterMap: Destination (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+Destination

get_authorization_details (**kw)

MWS GetAuthorizationDetails/2013-01-01 API call; quota=20 restore=2.00 Returns the status of a particular authorization and the total amount captured on the authorization.

Required: AmazonAuthorizationId

get_capture_details (**kw)

MWS GetCaptureDetails/2013-01-01 API call; quota=20 restore=2.00 Returns the status of a particular capture and the total amount refunded on the capture.

Required: AmazonCaptureId

get_cartinfo_service_status (*args, **kw)

MWS GetServiceStatus/2014-03-01 API call; quota=2 restore=300.00 Returns the operational status of the Cart Information API section.

get_carts (**kw)

MWS GetCarts/2014-03-01 API call; quota=15 restore=12.00 Returns shopping carts based on the CartId values that you specify.

Lists: CartIdList.CartId Required: CartIdList

get_competitive_pricing_for_asin (**kw)

MWS GetCompetitivePricingForASIN/2011-10-01 API call; quota=20 restore=10.00 Returns the current competitive pricing of a product, based on the ASINs and MarketplaceId that you specify.

Lists: ASINList.ASIN Required: MarketplaceId+ASINList

get_competitive_pricing_for_sku (**kw)

MWS GetCompetitivePricingForSKU/2011-10-01 API call; quota=20 restore=10.00 Returns the current competitive pricing of a product, based on the SellerSKUs and MarketplaceId that you specify.

Lists: SellerSKUList.SellerSKU Required: MarketplaceId+SellerSKUList

get_customerinfo_service_status (*args, **kw)

MWS GetServiceStatus/2014-03-01 API call; quota=2 restore=300.00 Returns the operational status of the Customer Information API section.

get_customers_for_customer_id (**kw)

MWS GetCustomersForCustomerId/2014-03-01 API call; quota=15 restore=12.00 Returns a list of customer accounts based on search criteria that you specify.

Lists: CustomerIdList.CustomerId Required: CustomerIdList

get_feed_submission_count (*args, **kw)

MWS GetFeedSubmissionCount/2009-01-01 API call; quota=10 restore=45.00 Returns a count of the feeds submitted in the previous 90 days.

Lists: FeedTypeList.Type, FeedProcessingStatusList.Status

get_feed_submission_list (*args, **kw)

MWS GetFeedSubmissionList/2009-01-01 API call; quota=10 restore=45.00 Returns a list of all feed

submissions submitted in the
previous 90 days.

Lists: FeedSubmissionIdList.Id, FeedTypeList.Type, FeedProcessingStatusList.Status

get_feed_submission_list_by_next_token (**kw)

MWS GetFeedSubmissionListByNextToken/2009-01-01 API call; quota=0 restore=0.00 Returns a list of feed submissions using the NextToken parameter.

Required: NextToken

get_feed_submission_result (**kw)

MWS GetFeedSubmissionResult/2009-01-01 API call; quota=15 restore=60.00 Returns the feed processing report.

Required: FeedSubmissionId

get_fulfillment_order (**kw)

MWS GetFulfillmentOrder/2010-10-01 API call; quota=30 restore=0.50 Returns a fulfillment order based on a specified

SellerFulfillmentOrderId.

Required: SellerFulfillmentOrderId

get_fulfillment_preview (**kw)

MWS GetFulfillmentPreview/2010-10-01 API call; quota=30 restore=0.50 Returns a list of fulfillment order previews based on items

and shipping speed categories that you specify.

ElementIterableMap: Address, Items (ResponseElement or anything iterable/dict-like) Required: Address+Items

get_inbound_service_status (*args, **kw)

MWS GetServiceStatus/2010-10-01 API call; quota=2 restore=300.00 Returns the operational status of the Fulfillment Inbound

Shipment API section.

get_inventory_service_status (*args, **kw)

MWS GetServiceStatus/2010-10-01 API call; quota=2 restore=300.00 Returns the operational status of the Fulfillment Inventory

API section.

get_last_updated_time_for_recommendations (**kw)

MWS GetLastUpdatedTimeForRecommendations/2013-04-01 API call; quota=5 restore=2.00 Checks whether there are active recommendations for each category

for the given marketplace, and if there are, returns the time when recommendations were last updated for each category.

Required: MarketplaceId

get_lowest_offer_listings_for_asin (**kw)

MWS GetLowestOfferListingsForASIN/2011-10-01 API call; quota=20 restore=5.00 Returns the lowest price offer listings for a specific

product by item condition and ASINs.

Lists: ASINList.ASIN Required: MarketplaceId+ASINList

get_lowest_offer_listings_for_sku (**kw)

MWS GetLowestOfferListingsForSKU/2011-10-01 API call; quota=20 restore=5.00 Returns the lowest price offer listings for a specific

product by item condition and SellerSKUs.

Lists: SellerSKUList.SellerSKU Required: MarketplaceId+SellerSKUList

get_matching_product (**kw)

MWS GetMatchingProduct/2011-10-01 API call; quota=20 restore=20.00 Returns a list of products and

their attributes, based on
a list of ASIN values that you specify.
Lists: ASINList.ASIN Required: MarketplaceId+ASINList

get_matching_product_for_id (**kw)

MWS GetMatchingProductForId/2011-10-01 API call; quota=20 restore=20.00 Returns a list of products and their attributes, based on
a list of Product IDs that you specify.
Lists: IdList.Id Required: MarketplaceId+IdType+IdList

get_my_price_for_asin (**kw)

MWS GetMyPriceForASIN/2011-10-01 API call; quota=20 restore=10.00 Returns pricing information for your own offer listings, based on ASIN.
Lists: ASINList.ASIN Required: MarketplaceId+ASINList

get_my_price_for_sku (**kw)

MWS GetMyPriceForSKU/2011-10-01 API call; quota=20 restore=10.00 Returns pricing information for your own offer listings, based on SellerSKU.
Lists: SellerSKUList.SellerSKU Required: MarketplaceId+SellerSKUList

get_offamazonpayments_service_status (*args, **kw)

MWS GetServiceStatus/2013-01-01 API call; quota=2 restore=300.00 Returns the operational status of the Off-Amazon Payments API section.

get_order (**kw)

MWS GetOrder/2013-09-01 API call; quota=6 restore=60.00 Returns an order for each AmazonOrderId that you specify.
Lists: AmazonOrderId.Id Required: AmazonOrderId

get_order_reference_details (**kw)

MWS GetOrderReferenceDetails/2013-01-01 API call; quota=20 restore=2.00 Returns details about the Order Reference object and its current state.
Required: AmazonOrderReferenceId

get_orders_service_status (*args, **kw)

MWS GetServiceStatus/2013-09-01 API call; quota=2 restore=300.00 Returns the operational status of the Orders API section.

get_outbound_service_status (*args, **kw)

MWS GetServiceStatus/2010-10-01 API call; quota=2 restore=300.00 Returns the operational status of the Fulfillment Outbound API section.

get_package_tracking_details (**kw)

MWS GetPackageTrackingDetails/2010-10-01 API call; quota=30 restore=0.50 Returns delivery tracking information for a package in
an outbound shipment for a Multi-Channel Fulfillment order.
Required: PackageNumber

get_product_categories_for_asin (**kw)

MWS GetProductCategoriesForASIN/2011-10-01 API call; quota=20 restore=20.00 Returns the product categories that an ASIN belongs to.
Required: MarketplaceId+ASIN

- get_product_categories_for_sku** (**kw)
MWS GetProductCategoriesForSKU/2011-10-01 API call; quota=20 restore=20.00 Returns the product categories that a SellerSKU belongs to.
Required: MarketplaceId+SellerSKU
- get_products_service_status** (*args, **kw)
MWS GetServiceStatus/2011-10-01 API call; quota=2 restore=300.00 Returns the operational status of the Products API section.
- get_recommendations_service_status** (*args, **kw)
MWS GetServiceStatus/2013-04-01 API call; quota=2 restore=300.00 Returns the operational status of the Recommendations API section.
- get_refund_details** (**kw)
MWS GetRefundDetails/2013-01-01 API call; quota=20 restore=2.00 Returns the status of a particular refund.
Required: AmazonRefundId
- get_report** (**kw)
MWS GetReport/2009-01-01 API call; quota=15 restore=60.00 Returns the contents of a report.
Required: ReportId
- get_report_count** (**kw)
MWS GetReportCount/2009-01-01 API call; quota=10 restore=45.00 Returns a count of the reports, created in the previous 90 days,
with a status of `_DONE_` and that are available for download.
Lists: ReportTypeList.Type Booleans: Acknowledged
- get_report_list** (**kw)
MWS GetReportList/2009-01-01 API call; quota=10 restore=60.00 Returns a list of reports that were created in the previous
90 days that match the query parameters.
Lists: ReportRequestIdList.Id, ReportTypeList.Type Booleans: Acknowledged
- get_report_list_by_next_token** (**kw)
MWS GetReportListByNextToken/2009-01-01 API call; quota=0 restore=0.00 Returns a list of reports using the NextToken, which
was supplied by a previous request to either GetReportListByNextToken or GetReportList,
where the value of HasNext was true in the previous call.
Required: NextToken
- get_report_request_count** (*args, **kw)
MWS GetReportRequestCount/2009-01-01 API call; quota=10 restore=45.00 Returns a count of report requests that have been submitted
to Amazon MWS for processing.
Lists: ReportTypeList.Type, ReportProcessingStatusList.Status
- get_report_request_list** (*args, **kw)
MWS GetReportRequestList/2009-01-01 API call; quota=10 restore=45.00 Returns a list of report requests that you can use to get the
ReportRequestId for a report.
Lists: ReportRequestIdList.Id, ReportTypeList.Type, ReportProcessingStatusList.Status
- get_report_request_list_by_next_token** (**kw)
MWS GetReportRequestListByNextToken/2009-01-01 API call; quota=0 restore=0.00 Returns a list of report requests using the NextToken,

which was supplied by a previous request to either `GetReportRequestListByNextToken` or `GetReportRequestList`, where the value of `HasNext` was true in that previous request.

Required: `NextToken`

get_report_schedule_count (**args*, ***kw*)

MWS `GetReportScheduleCount/2009-01-01` API call; quota=10 restore=45.00 Returns a count of order report requests that are scheduled to be submitted to Amazon MWS.

Lists: `ReportTypeList.Type`

get_report_schedule_list (**args*, ***kw*)

MWS `GetReportScheduleList/2009-01-01` API call; quota=10 restore=45.00 Returns a list of order report requests that are scheduled to be submitted to Amazon MWS for processing.

Lists: `ReportTypeList.Type`

get_report_schedule_list_by_next_token (***kw*)

MWS `GetReportScheduleListByNextToken/2009-01-01` API call; quota=0 restore=0.00 Returns a list of report requests using the `NextToken`, which was supplied by a previous request to either `GetReportScheduleListByNextToken` or `GetReportScheduleList`, where the value of `HasNext` was true in that previous request.

Required: `NextToken`

get_service_status (***kw*)

Instruct the user on how to get service status.

get_subscription (***kw*)

MWS `GetSubscription/2013-07-01` API call; quota=25 restore=0.50 Gets the subscription for the specified notification type and destination.

ElementIterMap: `Destination` (`ResponseElement` or anything iterable/dict-like) Required: `MarketplaceId+NotificationType+Destination`

get_subscriptions_service_status (**args*, ***kw*)

MWS `GetServiceStatus/2013-07-01` API call; quota=2 restore=300.00 Returns the operational status of the Subscriptions API section.

iter_call (*call*, **args*, ***kw*)

Pass a call name as the first argument and a generator is returned for the initial response and any continuation call responses made using the `NextToken`.

iter_response (*response*)

Pass a call's response as the initial argument and a generator is returned for the initial response and any continuation call responses made using the `NextToken`.

list_all_fulfillment_orders (**args*, ***kw*)

MWS `ListAllFulfillmentOrders/2010-10-01` API call; quota=30 restore=0.50 Returns a list of fulfillment orders fulfilled after (or at) a specified date or by fulfillment method.

list_all_fulfillment_orders_by_next_token (***kw*)

MWS `ListAllFulfillmentOrdersByNextToken/2010-10-01` API call; quota=30 restore=0.50 Returns the next page of inbound shipment items using the `NextToken` parameter.

Required: `NextToken`

list_carts (***kw*)

MWS `ListCarts/2014-03-01` API call; quota=15 restore=12.00 Returns a list of shopping carts in your Webstore that were last

updated during the time range that you specify.
Required: DateRangeStart

list_carts_by_next_token (**kw)

MWS ListCartsByNextToken/2014-03-01 API call; quota=50 restore=3.00 Returns the next page of shopping carts using the NextToken parameter.

Required: NextToken

list_customers (*args, **kw)

MWS ListCustomers/2014-03-01 API call; quota=15 restore=12.00 Returns a list of customer accounts based on search criteria that you specify.

list_customers_by_next_token (**kw)

MWS ListCustomersByNextToken/2014-03-01 API call; quota=50 restore=3.00 Returns the next page of customers using the NextToken parameter.

Required: NextToken

list_inbound_shipment_items (**kw)

MWS ListInboundShipmentItems/2010-10-01 API call; quota=30 restore=0.50 Returns a list of items in a specified inbound shipment, or a list of items that were updated within a specified time frame.

Required: ShipmentId OR LastUpdatedAfter+LastUpdatedBefore

list_inbound_shipment_items_by_next_token (**kw)

MWS ListInboundShipmentItemsByNextToken/2010-10-01 API call; quota=30 restore=0.50 Returns the next page of inbound shipment items using the NextToken parameter.

Required: NextToken

list_inbound_shipments (**kw)

MWS ListInboundShipments/2010-10-01 API call; quota=30 restore=0.50 Returns a list of inbound shipments based on criteria that you specify.

Lists: ShipmentIdList.Id, ShipmentStatusList.Status Some Required: ShipmentIdList, ShipmentStatusList

list_inbound_shipments_by_next_token (**kw)

MWS ListInboundShipmentsByNextToken/2010-10-01 API call; quota=30 restore=0.50 Returns the next page of inbound shipments using the NextToken parameter.

Required: NextToken

list_inventory_supply (**kw)

MWS ListInventorySupply/2010-10-01 API call; quota=30 restore=0.50 Returns information about the availability of a seller's inventory.

Lists: SellerSkus.member Required: SellerSkus OR QueryStartDateTime

list_inventory_supply_by_next_token (**kw)

MWS ListInventorySupplyByNextToken/2010-10-01 API call; quota=30 restore=0.50 Returns the next page of information about the availability of a seller's inventory using the NextToken parameter.

Required: NextToken

list_marketplace_participations (*args, **kw)

MWS ListMarketplaceParticipations/2011-07-01 API call; quota=15 restore=60.00 Returns a list of marketplaces that the seller submitting the request can sell in, and a list of participations that include seller-specific information in that marketplace.

list_marketplace_participations_by_next_token (**kw)

MWS ListMarketplaceParticipationsByNextToken/2011-07-01 API call; quota=15 restore=60.00 Returns the next page of marketplaces and participations using the NextToken value that was returned by your previous request to either ListMarketplaceParticipations or ListMarketplaceParticipationsByNextToken.
Required: NextToken

list_matching_products (**kw)

MWS ListMatchingProducts/2011-10-01 API call; quota=20 restore=20.00 Returns a list of products and their attributes, ordered by relevancy, based on a search query that you specify.
Required: MarketplaceId+Query

list_order_items (**kw)

MWS ListOrderItems/2013-09-01 API call; quota=30 restore=2.00 Returns order item information for an AmazonOrderId that you specify.
Required: AmazonOrderId

list_order_items_by_next_token (**kw)

MWS ListOrderItemsByNextToken/2013-09-01 API call; quota=30 restore=2.00 Returns the next page of order items using the NextToken value that was returned by your previous request to either ListOrderItems or ListOrderItemsByNextToken.
Required: NextToken

list_orders (**kw)

MWS ListOrders/2013-09-01 API call; quota=6 restore=60.00 Returns a list of orders created or updated during a time frame that you specify.
Lists: MarketplaceId.Id, OrderStatus.Status, FulfillmentChannel.Channel, PaymentMethod. ElementIterMap: OrderTotal, ShippingAddress, PaymentExecutionDetail (ResponseElement or anything iterable/dict-like) Either: CreatedAfter OR LastUpdatedBefore LastUpdatedBefore requires: LastUpdatedAfter Either: LastUpdatedAfter OR BuyerEmail OR SellerOrderId CreatedBefore requires: CreatedAfter Either: CreatedAfter OR LastUpdatedAfter Required: MarketplaceId Required: CreatedAfter OR LastUpdatedAfter

list_orders_by_next_token (**kw)

MWS ListOrdersByNextToken/2013-09-01 API call; quota=6 restore=60.00 Returns the next page of orders using the NextToken value that was returned by your previous request to either ListOrders or ListOrdersByNextToken.
Required: NextToken

list_recommendations (**kw)

MWS ListRecommendations/2013-04-01 API call; quota=5 restore=2.00 Returns your active recommendations for a specific category or for all categories for a specific marketplace.
Lists: CategoryQueryList.CategoryQuery Required: MarketplaceId

list_recommendations_by_next_token (**kw)

MWS ListRecommendationsByNextToken/2013-04-01 API call; quota=5 restore=2.00 Returns the next page of recommendations using the NextToken parameter.

Required: NextToken

list_registered_destinations (**kw)

MWS ListRegisteredDestinations/2013-07-01 API call; quota=25 restore=0.50 Lists all current destinations that you have registered.

Required: MarketplaceId

list_subscriptions (**kw)

MWS ListSubscriptions/2013-07-01 API call; quota=25 restore=0.50 Returns a list of all your current subscriptions.

Required: MarketplaceId

manage_report_schedule (**kw)

MWS ManageReportSchedule/2009-01-01 API call; quota=10 restore=45.00 Creates, updates, or deletes a report request schedule for a specified report type.

Required: ReportType+Schedule

method_for (name)

Return the MWS API method referred to in the argument. The named method can be in CamelCase or underlined_lower_case. This is the complement to MWSConnection.any_call.action

refund (**kw)

MWS Refund/2013-01-01 API call; quota=10 restore=1.00 Refunds a previously captured amount.

ElementIterableMap: RefundAmount (ResponseElement or anything iterable/dict-like) Required: AmazonCaptureId+RefundReferenceId+RefundAmount

register_destination (**kw)

MWS RegisterDestination/2013-07-01 API call; quota=25 restore=0.50 Specifies a new destination where you want to receive notifications.

ElementIterableMap: Destination (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+Destination

request_report (**kw)

MWS RequestReport/2009-01-01 API call; quota=15 restore=60.00 Creates a report request and submits the request to Amazon MWS.

Booleans: ReportOptions=ShowSalesChannel Lists: MarketplaceIdList.Id Required: ReportType

send_test_notification_to_destination (**kw)

MWS SendTestNotificationToDestination/2013-07-01 API call; quota=25 restore=0.50 Sends a test notification to an existing destination.

ElementIterableMap: Destination (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+Destination

set_order_reference_details (**kw)

MWS SetOrderReferenceDetails/2013-01-01 API call; quota=10 restore=1.00 Sets order reference details such as the order total and a description for the order.

ElementIterableMap: OrderReferenceAttributes (ResponseElement or anything iterable/dict-like) Required: AmazonOrderReferenceId+OrderReferenceAttributes

submit_feed (**kw)

MWS SubmitFeed/2009-01-01 API call; quota=15 restore=120.00 Uploads a feed for processing by Amazon MWS.

Lists: MarketplaceIdList.Id Required HTTP Body: FeedContent Booleans: PurgeAndReplace Required: FeedType

update_inbound_shipment (**kw)

MWS UpdateInboundShipment/2010-10-01 API call; quota=30 restore=0.50 Updates an existing inbound shipment. Amazon documentation

is ambiguous as to whether the InboundShipmentHeader and InboundShipmentItems arguments are required.

ElementIterlMap: InboundShipmentHeader, InboundShipmentItems (ResponseElement or anything iterable/dict-like) Required: ShipmentId

update_report_acknowledgements (**kw)

MWS UpdateReportAcknowledgements/2009-01-01 API call; quota=10 restore=45.00 Updates the acknowledged status of one or more reports.

Lists: ReportIdList.Id Booleans: Acknowledged Required: ReportIdList

update_subscription (**kw)

MWS UpdateSubscription/2013-07-01 API call; quota=25 restore=0.50 Updates the subscription for the specified notification type and destination.

ElementIterlMap: Subscription (ResponseElement or anything iterable/dict-like) Required: MarketplaceId+Subscription

boto.mws.exception

exception boto.mws.exception.**InvalidAddress** (*status, reason, body=None, *args*)
Invalid address.

exception boto.mws.exception.**InvalidParameter** (*status, reason, body=None, *args*)
One or more parameters in the request is invalid.

exception boto.mws.exception.**InvalidParameterValue** (*status, reason, body=None, *args*)
One or more parameter values in the request is invalid.

exception boto.mws.exception.**ResponseError** (*status, reason, body=None, *args*)
Undefined response error.

retry = False

class boto.mws.exception.**ResponseErrorFactory** (*scopes=None*)

exception boto.mws.exception.**RetriableResponseError** (*status, reason, body=None, *args*)

retry = True

boto.mws.response

class boto.mws.response.**AttributeSet** (*connection=None, name=None, parent=None, attrs=None*)

ItemDimensions = <Element_?/?_0x2ffe040>

ListPrice = <Element_?/?_0x2ffe040>

PackageDimensions = <Element_?/?_0x2ffe040>

SmallImage = <Element_?/?_0x2ffe040>

```
class boto.mws.response.AuthorizationDetails (connection=None, name=None, parent=None, attrs=None)

    AuthorizationAmount = <Element_?/?_0x2ffe040>
    AuthorizationFee = <Element_?/?_0x2ffe040>
    AuthorizationStatus = <Element_?/?_0x2ffe040>
    CapturedAmount = <Element_?/?_0x2ffe040>
class boto.mws.response.AuthorizeResult (connection=None, name=None, parent=None, attrs=None)

    AuthorizationDetails = <Element_?/?_0x2ffe040>
class boto.mws.response.CancelFeedSubmissionsResult (connection=None, name=None, parent=None, attrs=None)
class boto.mws.response.CancelReportRequestsResult (connection=None, name=None, parent=None, attrs=None)
class boto.mws.response.CaptureDetails (connection=None, name=None, parent=None, attrs=None)

    CaptureAmount = <Element_?/?_0x2ffe040>
    CaptureFee = <Element_?/?_0x2ffe040>
    CaptureStatus = <Element_?/?_0x2ffe040>
    RefundedAmount = <Element_?/?_0x2ffe040>
class boto.mws.response.CaptureResult (connection=None, name=None, parent=None, attrs=None)

    CaptureDetails = <Element_?/?_0x2ffe040>
class boto.mws.response.Cart (connection=None, name=None, parent=None, attrs=None)

    ActiveCartItem = <Element_?/?_0x2ffe040>
    SavedCartItem = <Element_?/?_0x2ffe040>
class boto.mws.response.CartItem (connection=None, name=None, parent=None, attrs=None)

    CurrentPrice = <Element_?/?_0x2ffe040>
    SalePrice = <Element_?/?_0x2ffe040>
class boto.mws.response.CompetitivePrice (connection=None, name=None, parent=None, attrs=None)

    Price = <Element_?/?_0x2ffe040>
class boto.mws.response.CompetitivePriceList (connection=None, name=None, parent=None, attrs=None)

    CompetitivePrice = <ElementList_?/?_0x49db0f0>
class boto.mws.response.CompetitivePricing (connection=None, name=None, parent=None, attrs=None)

    CompetitivePrices = <Element_?/?_0x2ffe040>
```

```

    NumberOfOfferListings = <SimpleList_?/?_0x30237f0>
    TradeInValue = <Element_?/?_0x2ffe040>
class boto.mws.response.ComplexAmount (connection=None, name=None, parent=None, at-
                                     trs=None)

    endElement (name, *args, **kw)
    startElement (name, *args, **kw)
class boto.mws.response.ComplexDimensions (connection=None, name=None, parent=None,
                                           attrs=None)

    endElement (name, *args, **kw)
    startElement (name, *args, **kw)
class boto.mws.response.ComplexMoney (connection=None, name=None, parent=None, at-
                                       trs=None)
class boto.mws.response.ComplexType
class boto.mws.response.ComplexWeight (connection=None, name=None, parent=None, at-
                                       trs=None)

    endElement (name, *args, **kw)
    startElement (name, *args, **kw)
class boto.mws.response.CreateInboundShipmentPlanResult (connection=None,
                                                         name=None,          par-
                                                         ent=None, attrs=None)

    InboundShipmentPlans = <MemberList_?/?_0x4d5f390>
class boto.mws.response.Customer (connection=None, name=None, parent=None, attrs=None)

    AssociatedMarketplaces = <Element_?/?_0x2ffe040>
    PrimaryContactInfo = <Element_?/?_0x2ffe040>
    ShippingAddressList = <Element_?/?_0x2ffe040>
class boto.mws.response.DeclarativeType (_hint=None, **kw)

    end (*args, **kw)
    setup (parent, name, *args, **kw)
    start (*args, **kw)
    teardown (*args, **kw)
class boto.mws.response.Destination (connection=None, name=None, parent=None, at-
                                     trs=None)

    AttributeList = <MemberList_?/?_0x4d5f390>
class boto.mws.response.Dimension
class boto.mws.response.Element (_hint=None, **kw)

    end (*args, **kw)

```

```
    start (*args, **kw)
class boto.mws.response.ElementList (*args, **kw)

    end (*args, **kw)
    start (*args, **kw)
class boto.mws.response.FeedSubmissionInfo (connection=None, name=None, parent=None,
                                             attrs=None)
class boto.mws.response.FulfillmentOrder (connection=None, name=None, parent=None,
                                             attrs=None)

    DestinationAddress = <Element_?/?_0x2ffe040>
    NotificationEmailList = <MemberList_?/?_0x4d5f390>
class boto.mws.response.FulfillmentPreview (connection=None, name=None, parent=None,
                                             attrs=None)

    EstimatedFees = <MemberList_?/?_0x4d5f390>
    EstimatedShippingWeight = <Element_?/?_0x2ffe040>
    FulfillmentPreviewShipments = <MemberList_?/?_0x4d5f390>
    UnfulfillablePreviewItems = <MemberList_?/?_0x4d5f390>
class boto.mws.response.FulfillmentPreviewItem (connection=None, name=None, parent=None,
                                                  attrs=None)

    EstimatedShippingWeight = <Element_?/?_0x2ffe040>
class boto.mws.response.GetAuthorizationDetailsResult (connection=None,
                                                       name=None, parent=None,
                                                       attrs=None)
class boto.mws.response.GetCaptureDetailsResult (connection=None, name=None, parent=None,
                                                  attrs=None)
class boto.mws.response.GetCartsResult (connection=None, name=None, parent=None,
                                         attrs=None)
class boto.mws.response.GetCompetitivePricingForASINResponse (*args, **kw)
class boto.mws.response.GetCompetitivePricingForSKUResponse (*args, **kw)
class boto.mws.response.GetCustomersForCustomerIdResult (connection=None,
                                                         name=None, parent=None,
                                                         attrs=None)
class boto.mws.response.GetFeedSubmissionCountResult (connection=None,
                                                       name=None, parent=None,
                                                       attrs=None)
class boto.mws.response.GetFeedSubmissionListResult (connection=None, name=None,
                                                      parent=None, attrs=None)

    FeedSubmissionInfo = <ElementList_?/?_0x49db0f0>
class boto.mws.response.GetFulfillmentOrderResult (connection=None, name=None,
                                                    parent=None, attrs=None)

    FulfillmentOrder = <Element_?/?_0x2ffe040>
```

```

FulfillmentOrderItem = <MemberList_?/?_0x4d5f390>
FulfillmentShipment = <MemberList_?/?_0x4d5f390>
class boto.mws.response.GetFulfillmentPreviewResult (connection=None, name=None,
parent=None, attrs=None)

FulfillmentPreviews = <MemberList_?/?_0x4d5f390>
class boto.mws.response.GetLowestOfferListingsForASINResponse (*args, **kw)
class boto.mws.response.GetLowestOfferListingsForSKUResponse (*args, **kw)
class boto.mws.response.GetMatchingProductForIdResponse (*args, **kw)
class boto.mws.response.GetMatchingProductForIdResult (connection=None,
name=None, parent=None,
attrs=None)

class boto.mws.response.GetMatchingProductResponse (*args, **kw)
class boto.mws.response.GetMyPriceForASINResponse (*args, **kw)
class boto.mws.response.GetMyPriceForSKUResponse (*args, **kw)
class boto.mws.response.GetOrderReferenceDetailsResult (connection=None,
name=None, parent=None,
attrs=None)

class boto.mws.response.GetOrderResult (connection=None, name=None, parent=None,
attrs=None)
class boto.mws.response.GetPackageTrackingDetailsResult (connection=None,
name=None, parent=None,
attrs=None)

ShipToAddress = <Element_?/?_0x2ffe040>
TrackingEvents = <MemberList_?/?_0x4d5f390>
class boto.mws.response.GetProductCategoriesForASINResult (connection=None,
name=None, parent=None,
attrs=None)
class boto.mws.response.GetProductCategoriesForSKUResult (connection=None,
name=None, parent=None,
attrs=None)
class boto.mws.response.GetProductCategoriesResult (connection=None, name=None,
parent=None, attrs=None)

Self = <ElementList_?/?_0x49db0f0>
class boto.mws.response.GetRefundDetails (connection=None, name=None, parent=None,
attrs=None)
class boto.mws.response.GetReportListResult (connection=None, name=None, parent=None,
attrs=None)

ReportInfo = <ElementList_?/?_0x49db0f0>
class boto.mws.response.GetReportRequestListResult (connection=None, name=None,
parent=None, attrs=None)

ReportRequestInfo = <ElementList_?/?_0x49db0f0>

```

```
class boto.mws.response.GetReportScheduleListResult (connection=None, name=None, parent=None, attrs=None)

class boto.mws.response.GetServiceStatusResult (connection=None, name=None, parent=None, attrs=None)

    Messages = <Element_?/?_0x2ffe040>

class boto.mws.response.GetSubscriptionResult (connection=None, name=None, parent=None, attrs=None)

    Subscription = <Element_?/?_0x2ffe040>

class boto.mws.response.Image (connection=None, name=None, parent=None, attrs=None)

class boto.mws.response.ItemAttributes (*args, **kw)

    Languages = <Element_?/?_0x2ffe040>

class boto.mws.response.ListAllFulfillmentOrdersResult (connection=None, name=None, parent=None, attrs=None)

    FulfillmentOrders = <MemberList_?/?_0x4d5f390>

class boto.mws.response.ListCartsResult (connection=None, name=None, parent=None, attrs=None)

    CartList = <Element_?/?_0x2ffe040>

class boto.mws.response.ListCustomersResult (connection=None, name=None, parent=None, attrs=None)

    CustomerList = <Element_?/?_0x2ffe040>

class boto.mws.response.ListInboundShipmentItemsResult (connection=None, name=None, parent=None, attrs=None)

    ItemData = <MemberList_?/?_0x4d5f390>

class boto.mws.response.ListInboundShipmentsResult (connection=None, name=None, parent=None, attrs=None)

    ShipmentData = <MemberList_?/?_0x4d5f390>

class boto.mws.response.ListInventorySupplyResult (connection=None, name=None, parent=None, attrs=None)

    InventorySupplyList = <MemberList_?/?_0x4d5f390>

class boto.mws.response.ListMarketplaceParticipationsResult (connection=None, name=None, parent=None, attrs=None)

    ListMarketplaces = <Element_?/?_0x2ffe040>

    ListParticipations = <Element_?/?_0x2ffe040>

class boto.mws.response.ListMatchingProductsResult (connection=None, name=None, parent=None, attrs=None)
```



```

    Products = <Element_?/?_0x2ffe040>
class boto.mws.response.ListOrderItemsResult (connection=None, name=None, parent=None,
                                                attrs=None)

    OrderItems = <Element_?/?_0x2ffe040>
class boto.mws.response.ListOrdersResult (connection=None, name=None, parent=None,
                                           attrs=None)

    Orders = <Element_?/?_0x2ffe040>
class boto.mws.response.ListRecommendationsResult (connection=None, name=None,
                                                    parent=None, attrs=None)

    ListingQualityRecommendations = <MemberList_?/?_0x4d5f390>
class boto.mws.response.ListRegisteredDestinationsResult (connection=None,
                                                           name=None, parent=None,
                                                           attrs=None)

    DestinationList = <MemberList_?/?_0x4d5f390>
class boto.mws.response.ListSubscriptionsResult (connection=None, name=None, parent=None,
                                                 attrs=None)

    SubscriptionList = <MemberList_?/?_0x4d5f390>
class boto.mws.response.LowestOfferListing (connection=None, name=None, parent=None,
                                             attrs=None)

    Price = <Element_?/?_0x2ffe040>
    Qualifiers = <Element_?/?_0x2ffe040>
class boto.mws.response.ManageReportScheduleResult (connection=None, name=None,
                                                    parent=None, attrs=None)

    ReportSchedule = <Element_?/?_0x2ffe040>
class boto.mws.response.MemberList (_member=None, _hint=None, *args, **kw)

    teardown (*args, **kw)
class boto.mws.response.Offer (connection=None, name=None, parent=None, attrs=None)

    BuyingPrice = <Element_?/?_0x2ffe040>
    RegularPrice = <Element_?/?_0x2ffe040>
class boto.mws.response.Order (connection=None, name=None, parent=None, attrs=None)

    OrderTotal = <Element_?/?_0x2ffe040>
    PaymentExecutionDetail = <Element_?/?_0x2ffe040>
    ShippingAddress = <Element_?/?_0x2ffe040>
class boto.mws.response.OrderItem (connection=None, name=None, parent=None, attrs=None)

    CODFee = <Element_?/?_0x2ffe040>

```

```
CODFeeDiscount = <Element_?/?_0x2ffe040>
GiftWrapPrice = <Element_?/?_0x2ffe040>
GiftWrapTax = <Element_?/?_0x2ffe040>
ItemPrice = <Element_?/?_0x2ffe040>
ItemTax = <Element_?/?_0x2ffe040>
PromotionDiscount = <Element_?/?_0x2ffe040>
PromotionIds = <SimpleList_?/?_0x30237f0>
ShippingDiscount = <Element_?/?_0x2ffe040>
ShippingPrice = <Element_?/?_0x2ffe040>
ShippingTax = <Element_?/?_0x2ffe040>
class boto.mws.response.OrderReferenceDetails (connection=None, name=None, parent=None, attrs=None)

    Buyer = <Element_?/?_0x2ffe040>
    Constraints = <ElementList_?/?_0x49db0f0>
    Destination = <Element_?/?_0x2ffe040>
    OrderReferenceStatus = <Element_?/?_0x2ffe040>
    OrderTotal = <Element_?/?_0x2ffe040>
    SellerOrderAttributes = <Element_?/?_0x2ffe040>
class boto.mws.response.Price (connection=None, name=None, parent=None, attrs=None)

    LandedPrice = <Element_?/?_0x2ffe040>
    ListingPrice = <Element_?/?_0x2ffe040>
    Shipping = <Element_?/?_0x2ffe040>
class boto.mws.response.Product (connection=None, name=None, parent=None, attrs=None)

    AttributeSets = <Element_?/?_0x2ffe040>
    CompetitivePricing = <ElementList_?/?_0x49db0f0>
    Identifiers = <Element_?/?_0x2ffe040>
    LowestOfferListings = <Element_?/?_0x2ffe040>
    Offers = <Element_?/?_0x2ffe040>
    Relationships = <Element_?/?_0x2ffe040>
    SalesRankings = <Element_?/?_0x2ffe040>
class boto.mws.response.ProductCategory (*args, **kw)
class boto.mws.response.ProductsBulkOperationResponse (*args, **kw)
class boto.mws.response.ProductsBulkOperationResult (connection=None, name=None, parent=None, attrs=None)

    Error = <Element_?/?_0x2ffe040>
```

```

    Product = <Element_?/?_0x2ffe040>
class boto.mws.response.RefundDetails (connection=None, name=None, parent=None, at-
trs=None)

    FeeRefunded = <Element_?/?_0x2ffe040>
    RefundAmount = <Element_?/?_0x2ffe040>
    RefundStatus = <Element_?/?_0x2ffe040>
class boto.mws.response.RefundResult (connection=None, name=None, parent=None, at-
trs=None)

    RefundDetails = <Element_?/?_0x2ffe040>
class boto.mws.response.ReportRequestInfo (connection=None, name=None, parent=None,
attrs=None)
class boto.mws.response.RequestReportResult (connection=None, name=None, par-
ent=None, attrs=None)

    ReportRequestInfo = <Element_?/?_0x2ffe040>
class boto.mws.response.Response (connection=None, name=None, parent=None, attrs=None)

    ResponseMetadata = <Element_?/?_0x2ffe040>
    startElement (name, *args, **kw)
class boto.mws.response.ResponseElement (connection=None, name=None, parent=None, at-
trs=None)

    connection
    endElement (name, *args, **kw)
    startElement (name, *args, **kw)
class boto.mws.response.ResponseFactory (scopes=None)

    element_factory (name, parent)
    find_element (action, suffix, parent)
    search_scopes (key)
class boto.mws.response.ResponseResultList (*args, **kw)
class boto.mws.response.SalesRank (connection=None, name=None, parent=None, attrs=None)
class boto.mws.response.SetOrderReferenceDetailsResult (connection=None,
name=None, parent=None,
attrs=None)

    OrderReferenceDetails = <Element_?/?_0x2ffe040>
class boto.mws.response.SimpleList (*args, **kw)

    end (name, value, *args, **kw)
    start (*args, **kw)

```

```
class boto.mws.response.SubmitFeedResult (connection=None, name=None, parent=None,  
                                           attrs=None)  
  
    FeedSubmissionInfo = <Element_?/?_0x2ffe040>  
class boto.mws.response.Subscription (connection=None, name=None, parent=None, at-  
                                       trs=None)  
  
    Destination = <Element_?/?_0x2ffe040>  
class boto.mws.response.UpdateReportAcknowledgementsResult (connection=None,  
                                                             name=None, par-  
                                                             ent=None, at-  
                                                             trs=None)  
  
class boto.mws.response.VariationRelationship (connection=None, name=None, par-  
                                               ent=None, attrs=None)  
  
    GemType = <SimpleList_?/?_0x30237f0>  
    Identifiers = <Element_?/?_0x2ffe040>  
    MaterialType = <SimpleList_?/?_0x30237f0>  
    OperatingSystem = <SimpleList_?/?_0x30237f0>  
boto.mws.response.strip_namespace (func)
```

4.138.5 pyami

boto.pyami

boto.pyami.bootstrap

class boto.pyami.bootstrap.**Bootstrap**

The Bootstrap class is instantiated and run as part of the PyAMI instance initialization process. The methods in this class will be run from the rc.local script of the instance and will be run as the root user.

The main purpose of this class is to make sure the boto distribution on the instance is the one required.

```
create_working_dir ()  
fetch_s3_file (s3_file)  
load_boto ()  
load_packages ()  
main ()  
write_metadata ()
```

boto.pyami.config

class boto.pyami.config.**Config** (*path=None, fp=None, do_load=True*)

```
dump ()  
dump_safe (fp=None)  
dump_to_sdb (domain_name, item_name)
```

```

get (section, name, default=None)
get_instance (name, default=None)
get_user (name, default=None)
get_value (section, name, default=None)
getbool (section, name, default=False)
getfloat (section, name, default=0.0)
getint (section, name, default=0)
getint_user (name, default=0)
has_option (*args, **kwargs)
load_credential_file (path)
    Load a credential file as is setup like the Java utilities
load_from_path (path)
load_from_sdb (domain_name, item_name)
save_option (path, section, option, value)
    Write the specified Section.Option to the config file specified by path. Replace any previous value. If the
    path doesn't exist, create it. Also add the option the the in-memory config.
save_system_option (section, option, value)
save_user_option (section, option, value)
setbool (section, name, value)

```

boto.pyami.copybot

```
class boto.pyami.copybot.CopyBot
```

```

copy_bucket_acl ()
copy_key_acl (src, dst)
copy_keys ()
copy_log ()
main ()

```

boto.pyami.installers

```
class boto.pyami.installers.Installer (config_file=None)
    Abstract base class for installers

add_cron (name, minute, hour, mday, month, wday, who, command, env=None)
    Add an entry to the system crontab.

add_env (key, value)
    Add an environemnt variable

add_init_script (file)
    Add this file to the init.d directory

```

```
install ()
    Do whatever is necessary to “install” the package.

start (service_name)
    Start a service.

stop (service_name)
    Stop a service.
```

boto.pyami.installers.ubuntu

boto.pyami.installers.ubuntu.apache

```
class boto.pyami.installers.ubuntu.apache.Apache (config_file=None)
    Install apache2, mod_python, and libapache2-svn

    install ()
        This is the only method you need to override

    main ()
```

boto.pyami.installers.ubuntu.ebs

boto.pyami.installers.ubuntu.installer

```
class boto.pyami.installers.ubuntu.installer.Installer (config_file=None)
    Base Installer class for Ubuntu-based AMI's

    add_cron (name, command, minute='*', hour='*', mday='*', month='*', wday='*', who='root',
              env=None)
        Write a file to /etc/cron.d to schedule a command env is a dict containing environment variables you
        want to set in the file name will be used as the name of the file

    add_env (key, value)
        Add an environemnt variable For Ubuntu, the best place is /etc/environment. Values placed here do not
        need to be exported.

    add_init_script (file, name)
        Add this file to the init.d directory

    create_user (user)
        Create a user on the local system

    install ()
        This is the only method you need to override

    start (service_name)
        Start a service.

    stop (service_name)
        Stop a service.
```

boto.pyami.installers.ubuntu.mysql

This installer will install mysql-server on an Ubuntu machine. In addition to the normal installation done by apt-get, it will also configure the new MySQL server to store it's data files in a different location. By default, this is /mnt but that can be configured in the [MySQL] section of the boto config file passed to the instance.

```
class boto.pyami.installers.ubuntu.mysql.MySQL (config_file=None)
```

```
    change_data_dir (password=None)
```

```
    install ()
```

```
        This is the only method you need to override
```

```
    main ()
```

boto.pyami.installers.ubuntu.trac

```
class boto.pyami.installers.ubuntu.trac.Trac (config_file=None)
```

```
    Install Trac and DAV-SVN Sets up a Vhost pointing to [Trac]->home Using the config parameter [Trac]->hostname Sets up a trac environment for every directory found under [Trac]->data_dir
```

```
[Trac] name = My Foo Server hostname = trac.foo.com home = /mnt/sites/trac data_dir = /mnt/trac svn_dir = /mnt/subversion server_admin = root@foo.com sdb_auth_domain = users # Optional SSLCertificateFile = /mnt/ssl/foo.crt SSLCertificateKeyFile = /mnt/ssl/foo.key SSLCertificateChainFile = /mnt/ssl/FooCA.crt
```

```
    install ()
```

```
        This is the only method you need to override
```

```
    main ()
```

```
    setup_vhost ()
```

boto.pyami.launch_ami

```
boto.pyami.launch_ami.main ()
```

```
boto.pyami.launch_ami.usage ()
```

boto.pyami.scriptbase

```
class boto.pyami.scriptbase.ScriptBase (config_file=None)
```

```
    main ()
```

```
    mkdir (path)
```

```
    notify (subject, body="")
```

```
    run (command, notify=True, exit_on_error=False, cwd=None)
```

```
    umount (path)
```

boto.pyami.startup

```
class boto.pyami.startup.Startup (config_file=None)
```

```
    main ()
```

```
    run_scripts ()
```

4.138.6 RDS

boto.rds

```
class boto.rds.RDSConnection(aws_access_key_id=None, aws_secret_access_key=None,  
is_secure=True, port=None, proxy=None, proxy_port=None,  
proxy_user=None, proxy_pass=None, debug=0,  
https_connection_factory=None, region=None, path='/', security_token=None, validate_certs=True, profile_name=None)
```

```
APIVersion = '2013-05-15'
```

```
DefaultRegionEndpoint = 'rds.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
authorize_dbsecurity_group(group_name, cidr_ip=None, ec2_security_group_name=None,  
ec2_security_group_owner_id=None)
```

Add a new rule to an existing security group. You need to pass in either `src_security_group_name` and `src_security_group_owner_id` OR a CIDR block but not both.

Parameters

- **group_name** (*string*) – The name of the security group you are adding the rule to.
- **ec2_security_group_name** (*string*) – The name of the EC2 security group you are granting access to.
- **ec2_security_group_owner_id** (*string*) – The ID of the owner of the EC2 security group you are granting access to.
- **cidr_ip** (*string*) – The CIDR block you are providing access to. See http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Return type `bool`

Returns True if successful.

```
copy_dbsnapshot(source_snapshot_id, target_snapshot_id)
```

Copies the specified DBSnapshot.

Parameters

- **source_snapshot_id** (*string*) – The identifier for the source DB snapshot.
- **target_snapshot_id** (*string*) – The identifier for the copied snapshot.

Return type `boto.rds.dbsnapshot.DBSnapshot`

Returns The newly created DBSnapshot.

```
create_db_subnet_group(name, desc, subnet_ids)
```

Create a new Database Subnet Group.

Parameters

- **name** (*string*) – The identifier for the `db_subnet_group`
- **desc** (*string*) – A description of the `db_subnet_group`
- **subnets** – A list of the subnet identifiers to include in the `db_subnet_group`

Return type `:class:`boto.rds.dbsubnetgroup.DBSubnetGroup`

Returns the created `db_subnet_group`


```
create_dbinstance (id, allocated_storage, instance_class, master_username, master_password, port=3306, engine='MySQL5.1', db_name=None, param_group=None, security_groups=None, availability_zone=None, preferred_maintenance_window=None, backup_retention_period=None, preferred_backup_window=None, multi_az=False, engine_version=None, auto_minor_version_upgrade=True, character_set_name=None, db_subnet_group_name=None, license_model=None, option_group_name=None, iops=None, vpc_security_groups=None)
```

Create a new DBInstance.

Parameters

- **id** (*str*) – Unique identifier for the new instance. Must contain 1-63 alphanumeric characters. First character must be a letter. May not end with a hyphen or contain two consecutive hyphens
- **allocated_storage** (*int*) – Initially allocated storage size, in GBs. Valid values are depending on the engine value.
 - MySQL = 5–3072
 - oracle-se1 = 10–3072
 - oracle-se = 10–3072
 - oracle-ee = 10–3072
 - sqlserver-ee = 200–1024
 - sqlserver-se = 200–1024
 - sqlserver-ex = 30–1024
 - sqlserver-web = 30–1024
 - postgres = 5–3072
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Valid values are:
 - db.t1.micro
 - db.m1.small
 - db.m1.medium
 - db.m1.large
 - db.m1.xlarge
 - db.m2.xlarge
 - db.m2.2xlarge
 - db.m2.4xlarge
- **engine** (*str*) – Name of database engine. Defaults to MySQL but can be;
 - MySQL
 - oracle-se1
 - oracle-se
 - oracle-ee
 - sqlserver-ee
 - sqlserver-se

- sqlserver-ex
- sqlserver-web
- postgres
- **master_username** (*str*) – Name of master user for the DBInstance.
 - MySQL must be: - 1–16 alphanumeric characters - first character must be a letter - cannot be a reserved MySQL word
 - Oracle must be: - 1–30 alphanumeric characters - first character must be a letter - cannot be a reserved Oracle word
 - SQL Server must be: - 1–128 alphanumeric characters - first character must be a letter - cannot be a reserved SQL Server word
- **master_password** (*str*) – Password of master user for the DBInstance.
 - MySQL must be 8–41 alphanumeric characters
 - Oracle must be 8–30 alphanumeric characters
 - SQL Server must be 8–128 alphanumeric characters.
- **port** (*int*) – Port number on which database accepts connections. Valid values [1115-65535].
 - MySQL defaults to 3306
 - Oracle defaults to 1521
 - SQL Server defaults to 1433 and `_cannot_` be 1434, 3389, 47001, 49152, and 49152 through 49156.
 - PostgreSQL defaults to 5432
- **db_name** (*str*) –
 - MySQL: Name of a database to create when the DBInstance is created. Default is to create no databases.
Must contain 1–64 alphanumeric characters and cannot be a reserved MySQL word.
 - Oracle: The Oracle System ID (SID) of the created DB instances. Default is ORCL. Cannot be longer than 8 characters.
 - SQL Server: Not applicable and must be None.
 - PostgreSQL: Name of a database to create when the DBInstance is created. Default is to create no databases.
Must contain 1–63 alphanumeric characters. Must begin with a letter or an underscore. Subsequent characters can be letters, underscores, or digits (0-9) and cannot be a reserved PostgreSQL word.
- **param_group** (*str or ParameterGroup object*) – Name of DBParameterGroup or ParameterGroup instance to associate with this DBInstance. If no groups are specified no parameter groups will be used.
- **security_groups** (*list of str or list of DBSecurityGroup objects*) – List of names of DBSecurityGroup to authorize on this DBInstance.

- **availability_zone** (*str*) – Name of the availability zone to place DBInstance into.
- **preferred_maintenance_window** (*str*) – The weekly time range (in UTC) during which maintenance can occur. Default is Sun:05:00-Sun:09:00
- **backup_retention_period** (*int*) – The number of days for which automated backups are retained. Setting this to zero disables automated backups.
- **preferred_backup_window** (*str*) – The daily time range during which automated backups are created (if enabled). Must be in h24:mi-hh24:mi format (UTC).
- **multi_az** (*bool*) – If True, specifies the DB Instance will be deployed in multiple availability zones.

For Microsoft SQL Server, must be set to false. You cannot set the Availability-Zone parameter if the MultiAZ parameter is set to true.

- **engine_version** (*str*) – The version number of the database engine to use.
 - MySQL format example: 5.1.42
 - Oracle format example: 11.2.0.2.v2
 - SQL Server format example: 10.50.2789.0.v1
 - PostgreSQL format example: 9.3
- **auto_minor_version_upgrade** (*bool*) – Indicates that minor engine upgrades will be applied automatically to the Read Replica during the maintenance window. Default is True.
- **character_set_name** (*str*) – For supported engines, indicates that the DB Instance should be associated with the specified CharacterSet.
- **db_subnet_group_name** (*str*) – A DB Subnet Group to associate with this DB Instance. If there is no DB Subnet Group, then it is a non-VPC DB instance.
- **license_model** (*str*) – License model information for this DB Instance.
Valid values are; - license-included - bring-your-own-license - general-public-license
All license types are not supported on all engines.
- **option_group_name** (*str*) – Indicates that the DB Instance should be associated with the specified option group.
- **iops** (*int*) – The amount of IOPS (input/output operations per second) to Provisioned for the DB Instance. Can be modified at a later date.
Must scale linearly. For every 1000 IOPS provision, you must allocated 100 GB of storage space. This scales up to 1 TB / 10 000 IOPS for MySQL and Oracle. MSSQL is limited to 700 GB / 7 000 IOPS.
If you specify a value, it must be at least 1000 IOPS and you must allocate 100 GB of storage.
- **vpc_security_groups** (*list of str or a VPCSecurityGroupMembership object*) – List of VPC security group ids or a list of VPCSecurityGroupMembership objects this DBInstance should be a member of

Return type *boto.rds.dbinstance.DBInstance*

Returns The new db instance.

create_dbinstance_read_replica (*id, source_id, instance_class=None, port=3306, availability_zone=None, auto_minor_version_upgrade=None*)

Create a new DBInstance Read Replica.

Parameters

- **id** (*str*) – Unique identifier for the new instance. Must contain 1-63 alphanumeric characters. First character must be a letter. May not end with a hyphen or contain two consecutive hyphens
- **source_id** (*str*) – Unique identifier for the DB Instance for which this DB Instance will act as a Read Replica.
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Default is to inherit from the source DB Instance.

Valid values are:

- db.m1.small
 - db.m1.large
 - db.m1.xlarge
 - db.m2.xlarge
 - db.m2.2xlarge
 - db.m2.4xlarge
- **port** (*int*) – Port number on which database accepts connections. Default is to inherit from source DB Instance. Valid values [1115-65535]. Defaults to 3306.
 - **availability_zone** (*str*) – Name of the availability zone to place DBInstance into.
 - **auto_minor_version_upgrade** (*bool*) – Indicates that minor engine upgrades will be applied automatically to the Read Replica during the maintenance window. Default is to inherit this value from the source DB Instance.

Return type *boto.rds.dbinstance.DBInstance*

Returns The new db instance.

create_dbsecurity_group (*name, description=None*)

Create a new security group for your account. This will create the security group within the region you are currently connected to.

Parameters

- **name** (*string*) – The name of the new security group
- **description** (*string*) – The description of the new security group

Return type *boto.rds.dbsecuritygroup.DBSecurityGroup*

Returns The newly created DBSecurityGroup

create_dbsnapshot (*snapshot_id, dbinstance_id*)

Create a new DB snapshot.

Parameters

- **snapshot_id** (*string*) – The identifier for the DBSnapshot
- **dbinstance_id** (*string*) – The source identifier for the RDS instance from which the snapshot is created.

Return type `boto.rds.dbsnapshot.DBSnapshot`

Returns The newly created DBSnapshot

create_option_group (*name, engine_name, major_engine_version, description=None*)

Create a new option group for your account. This will create the option group within the region you are currently connected to.

Parameters

- **name** (*string*) – The name of the new option group
- **engine_name** (*string*) – Specifies the name of the engine that this option group should be associated with.
- **major_engine_version** (*string*) – Specifies the major version of the engine that this option group should be associated with.
- **description** (*string*) – The description of the new option group

Return type `boto.rds.optiongroup.OptionGroup`

Returns The newly created OptionGroup

create_parameter_group (*name, engine='MySQL5.1', description=""*)

Create a new dbparameter group for your account.

Parameters

- **name** (*string*) – The name of the new dbparameter group
- **engine** (*str*) – Name of database engine.
- **description** (*string*) – The description of the new dbparameter group

Return type `boto.rds.parametergroup.ParameterGroup`

Returns The newly created ParameterGroup

delete_db_subnet_group (*name*)

Delete a Database Subnet Group.

Parameters **name** (*string*) – The identifier of the db_subnet_group to delete

Return type `boto.rds.dbsubnetgroup.DBSubnetGroup`

Returns The deleted db_subnet_group.

delete_dbinstance (*id, skip_final_snapshot=False, final_snapshot_id=""*)

Delete an existing DBInstance.

Parameters

- **id** (*str*) – Unique identifier for the new instance.
- **skip_final_snapshot** (*bool*) – This parameter determines whether a final db snapshot is created before the instance is deleted. If True, no snapshot is created. If False, a snapshot is created before deleting the instance.
- **final_snapshot_id** (*str*) – If a final snapshot is requested, this is the identifier used for that snapshot.

Return type `boto.rds.dbinstance.DBInstance`

Returns The deleted db instance.

delete_dbsecurity_group (*name*)

Delete a DBSecurityGroup from your account.

Parameters **key_name** (*string*) – The name of the DBSecurityGroup to delete

delete_dbsnapshot (*identifier*)

Delete a DBSnapshot

Parameters **identifier** (*string*) – The identifier of the DBSnapshot to delete

delete_option_group (*name*)

Delete an OptionGroup from your account.

Parameters **key_name** (*string*) – The name of the OptionGroup to delete

delete_parameter_group (*name*)

Delete a ParameterGroup from your account.

Parameters **key_name** (*string*) – The name of the ParameterGroup to delete

describe_option_group_options (*engine_name=None, major_engine_version=None, max_records=100, marker=None*)

Describes the available option group options.

Parameters

- **engine_name** (*str*) – Filters the list of option groups to only include groups associated with a specific database engine.
- **major_engine_version** (*datetime*) – Filters the list of option groups to only include groups associated with a specific database engine version. If specified, then **engine_name** must also be specified.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of class:*boto.rds.optiongroup.Option*

describe_option_groups (*name=None, engine_name=None, major_engine_version=None, max_records=100, marker=None*)

Describes the available option groups.

Parameters

- **name** (*str*) – The name of the option group to describe. Cannot be supplied together with **engine_name** or **major_engine_version**.
- **engine_name** (*str*) – Filters the list of option groups to only include groups associated with a specific database engine.
- **major_engine_version** (*datetime*) – Filters the list of option groups to only include groups associated with a specific database engine version. If specified, then **engine_name** must also be specified.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of class:*boto.rds.optiongroup.OptionGroup*

get_all_db_subnet_groups (*name=None, max_records=None, marker=None*)

Retrieve all the DBSubnetGroups in your account.

Parameters

- **name** (*str*) – DBSubnetGroup name. If supplied, only information about this DBSubnetGroup will be returned. Otherwise, info about all DBSubnetGroups will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a Token will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of `boto.rds.dbsubnetgroup.DBSubnetGroup`

get_all_dbinstances (*instance_id=None, max_records=None, marker=None*)

Retrieve all the DBInstances in your account.

Parameters

- **instance_id** (*str*) – DB Instance identifier. If supplied, only information this instance will be returned. Otherwise, info about all DB Instances will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of `boto.rds.dbinstance.DBInstance`

get_all_dbparameter_groups (*groupname=None, max_records=None, marker=None*)

Get all parameter groups associated with your account in a region.

Parameters

- **groupname** (*str*) – The name of the DBParameter group to retrieve. If not provided, all DBParameter groups will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of `boto.ec2.parametergroup.ParameterGroup`

get_all_dbparameters (*groupname, source=None, max_records=None, marker=None*)

Get all parameters associated with a ParameterGroup

Parameters

- **groupname** (*str*) – The name of the DBParameter group to retrieve.
- **source** (*str*) – Specifies which parameters to return. If not specified, all parameters will be returned. Valid values are: `usersystemengine-default`
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type `boto.ec2.parametergroup.ParameterGroup`

Returns The ParameterGroup

get_all_dbsecurity_groups (*groupname=None, max_records=None, marker=None*)

Get all security groups associated with your account in a region.

Parameters

- **groupnames** (*list*) – A list of the names of security groups to retrieve. If not provided, all security groups will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of *boto.rds.dbsecuritygroup.DBSecurityGroup*

get_all_dbsnapshots (*snapshot_id=None, instance_id=None, max_records=None, marker=None*)

Get information about DB Snapshots.

Parameters

- **snapshot_id** (*str*) – The unique identifier of an RDS snapshot. If not provided, all RDS snapshots will be returned.
- **instance_id** (*str*) – The identifier of a DBInstance. If provided, only the DBSnapshots related to that instance will be returned. If not provided, all RDS snapshots will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.
- **marker** (*str*) – The marker provided by a previous request.

Return type *list*

Returns A list of *boto.rds.dbsnapshot.DBSnapshot*

get_all_events (*source_identifier=None, source_type=None, start_time=None, end_time=None, max_records=None, marker=None*)

Get information about events related to your DBInstances, DBSecurityGroups and DBParameterGroups.

Parameters

- **source_identifier** (*str*) – If supplied, the events returned will be limited to those that apply to the identified source. The value of this parameter depends on the value of source_type. If neither parameter is specified, all events in the time span will be returned.
- **source_type** (*str*) – Specifies how the source_identifier should be interpreted. Valid values are: b-instance | db-security-group | db-parameter-group | db-snapshot
- **start_time** (*datetime*) – The beginning of the time interval for events. If not supplied, all available events will be returned.
- **end_time** (*datetime*) – The ending of the time interval for events. If not supplied, all available events will be returned.
- **max_records** (*int*) – The maximum number of records to be returned. If more results are available, a MoreToken will be returned in the response that can be used to retrieve additional records. Default is 100.

- **marker** (*str*) – The marker provided by a previous request.

Return type list

Returns A list of class:*boto.rds.event.Event*

get_all_logs (*dbinstance_id*, *max_records=None*, *marker=None*, *file_size=None*, *filename_contains=None*, *file_last_written=None*)

Get all log files

Parameters

- **instance_id** (*str*) – The identifier of a DBInstance.
- **max_records** (*int*) – Number of log file names to return.
- **marker** (*str*) – The marker provided by a previous request.
- **file_size** – Filter results to files large than this size in bytes.
- **filename_contains** – Filter results to files with filename containing this string
- **file_last_written** – Filter results to files written after this time (POSIX timestamp)

File_size int

Filename_contains str

File_last_written int

Return type list

Returns A list of *boto.rds.logfile.LogFile*

get_log_file (*dbinstance_id*, *log_file_name*, *marker=None*, *number_of_lines=None*, *max_records=None*)

Download a log file from RDS

Parameters

- **instance_id** (*str*) – The identifier of a DBInstance.
- **log_file_name** (*str*) – The name of the log file to retrieve
- **marker** (*str*) – A marker returned from a previous call to this method, or 0 to indicate the start of file. If no marker is specified, this will fetch log lines from the end of file instead.
- **marker** – The maximum number of lines to be returned.

modify_db_subnet_group (*name*, *description=None*, *subnet_ids=None*)

Modify a parameter group for your account.

Parameters

- **name** (*string*) – The name of the new parameter group
- **parameters** (list of *boto.rds.parametergroup.Parameter*) – The new parameters

Return type *boto.rds.parametergroup.ParameterGroup*

Returns The newly created ParameterGroup

modify_dbinstance (*id*, *param_group=None*, *security_groups=None*, *preferred_maintenance_window=None*, *master_password=None*, *allocated_storage=None*, *instance_class=None*, *backup_retention_period=None*, *preferred_backup_window=None*, *multi_az=False*, *apply_immediately=False*, *iops=None*, *vpc_security_groups=None*, *new_instance_id=None*)

Modify an existing DBInstance.

Parameters

- **id** (*str*) – Unique identifier for the new instance.
- **param_group** (*str* or *ParameterGroup* object) – Name of DBParameterGroup or ParameterGroup instance to associate with this DBInstance. If no groups are specified no parameter groups will be used.
- **security_groups** (*list of str* or *list of DBSecurityGroup* objects) – List of names of DBSecurityGroup to authorize on this DBInstance.
- **preferred_maintenance_window** (*str*) – The weekly time range (in UTC) during which maintenance can occur. Default is Sun:05:00-Sun:09:00
- **master_password** (*str*) – Password of master user for the DBInstance. Must be 4-15 alphanumeric characters.
- **allocated_storage** (*int*) – The new allocated storage size, in GBs. Valid values are [5-1024]
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Changes will be applied at next maintenance window unless *apply_immediately* is True.
Valid values are:
 - db.m1.small
 - db.m1.large
 - db.m1.xlarge
 - db.m2.xlarge
 - db.m2.2xlarge
 - db.m2.4xlarge
- **apply_immediately** (*bool*) – If true, the modifications will be applied as soon as possible rather than waiting for the next preferred maintenance window.
- **backup_retention_period** (*int*) – The number of days for which automated backups are retained. Setting this to zero disables automated backups.
- **preferred_backup_window** (*str*) – The daily time range during which automated backups are created (if enabled). Must be in h24:mi-hh24:mi format (UTC).
- **multi_az** (*bool*) – If True, specifies the DB Instance will be deployed in multiple availability zones.
- **iops** (*int*) – The amount of IOPS (input/output operations per second) to Provisioned for the DB Instance. Can be modified at a later date.

Must scale linearly. For every 1000 IOPS provision, you must allocated 100 GB of storage space. This scales up to 1 TB / 10 000 IOPS for MySQL and Oracle. MSSQL is limited to 700 GB / 7 000 IOPS.

If you specify a value, it must be at least 1000 IOPS and you must allocate 100 GB of storage.

- **vpc_security_groups** (*list of str or a VPCSecurityGroupMembership object*) – List of VPC security group ids or a VPCSecurityGroupMembership object this DBInstance should be a member of
- **new_instance_id** (*str*) – New name to rename the DBInstance to.

Return type *boto.rds.dbinstance.DBInstance*

Returns The modified db instance.

modify_parameter_group (*name, parameters=None*)

Modify a ParameterGroup for your account.

Parameters

- **name** (*string*) – The name of the new ParameterGroup
- **parameters** (*list of boto.rds.parametergroup.Parameter*) – The new parameters

Return type *boto.rds.parametergroup.ParameterGroup*

Returns The newly created ParameterGroup

promote_read_replica (*id, backup_retention_period=None, preferred_backup_window=None*)

Promote a Read Replica to a standalone DB Instance.

Parameters

- **id** (*str*) – Unique identifier for the new instance. Must contain 1-63 alphanumeric characters. First character must be a letter. May not end with a hyphen or contain two consecutive hyphens
- **backup_retention_period** (*int*) – The number of days for which automated backups are retained. Setting this to zero disables automated backups.
- **preferred_backup_window** (*str*) – The daily time range during which automated backups are created (if enabled). Must be in h24:mi-hh24:mi format (UTC).

Return type *boto.rds.dbinstance.DBInstance*

Returns The new db instance.

reboot_dbinstance (*id*)

Reboot DBInstance.

Parameters **id** (*str*) – Unique identifier of the instance.

Return type *boto.rds.dbinstance.DBInstance*

Returns The rebooting db instance.

reset_parameter_group (*name, reset_all_params=False, parameters=None*)

Resets some or all of the parameters of a ParameterGroup to the default value

Parameters

- **key_name** (*string*) – The name of the ParameterGroup to reset

- **parameters** (list of *boto.rds.parametergroup.Parameter*) – The parameters to reset. If not supplied, all parameters will be reset.

restore_dbinstance_from_dbsnapshot (*identifier, instance_id, instance_class, port=None, availability_zone=None, multi_az=None, auto_minor_version_upgrade=None, db_subnet_group_name=None*)

Create a new DBInstance from a DB snapshot.

Parameters

- **identifier** (*string*) – The identifier for the DBSnapshot
- **instance_id** (*string*) – The source identifier for the RDS instance from which the snapshot is created.
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Valid values are: db.m1.small | db.m1.large | db.m1.xlarge | db.m2.2xlarge | db.m2.4xlarge
- **port** (*int*) – Port number on which database accepts connections. Valid values [1115-65535]. Defaults to 3306.
- **availability_zone** (*str*) – Name of the availability zone to place DBInstance into.
- **multi_az** (*bool*) – If True, specifies the DB Instance will be deployed in multiple availability zones. Default is the API default.
- **auto_minor_version_upgrade** (*bool*) – Indicates that minor engine upgrades will be applied automatically to the Read Replica during the maintenance window. Default is the API default.
- **db_subnet_group_name** (*str*) – A DB Subnet Group to associate with this DB Instance. If there is no DB Subnet Group, then it is a non-VPC DB instance.

Return type *boto.rds.dbinstance.DBInstance*

Returns The newly created DBInstance

restore_dbinstance_from_point_in_time (*source_instance_id, target_instance_id, use_latest=False, restore_time=None, dbinstance_class=None, port=None, availability_zone=None, db_subnet_group_name=None*)

Create a new DBInstance from a point in time.

Parameters

- **source_instance_id** (*string*) – The identifier for the source DBInstance.
- **target_instance_id** (*string*) – The identifier of the new DBInstance.
- **use_latest** (*bool*) – If True, the latest snapshot available will be used.
- **restore_time** (*datetime*) – The date and time to restore from. Only used if use_latest is False.
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Valid values are: db.m1.small | db.m1.large | db.m1.xlarge | db.m2.2xlarge | db.m2.4xlarge
- **port** (*int*) – Port number on which database accepts connections. Valid values [1115-65535]. Defaults to 3306.

- **availability_zone** (*str*) – Name of the availability zone to place DBInstance into.
- **db_subnet_group_name** (*str*) – A DB Subnet Group to associate with this DB Instance. If there is no DB Subnet Group, then it is a non-VPC DB instance.

Return type *boto.rds.dbinstance.DBInstance*

Returns The newly created DBInstance

revoke_dbsecurity_group (*group_name*, *ec2_security_group_name=None*,
ec2_security_group_owner_id=None, *cidr_ip=None*)

Remove an existing rule from an existing security group. You need to pass in either *ec2_security_group_name* and *ec2_security_group_owner_id* OR a CIDR block.

Parameters

- **group_name** (*string*) – The name of the security group you are removing the rule from.
- **ec2_security_group_name** (*string*) – The name of the EC2 security group from which you are removing access.
- **ec2_security_group_owner_id** (*string*) – The ID of the owner of the EC2 security from which you are removing access.
- **cidr_ip** (*string*) – The CIDR block from which you are removing access. See http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Return type *bool*

Returns True if successful.

revoke_security_group (*group_name*, *ec2_security_group_name=None*,
ec2_security_group_owner_id=None, *cidr_ip=None*)

Remove an existing rule from an existing security group. You need to pass in either *ec2_security_group_name* and *ec2_security_group_owner_id* OR a CIDR block.

Parameters

- **group_name** (*string*) – The name of the security group you are removing the rule from.
- **ec2_security_group_name** (*string*) – The name of the EC2 security group from which you are removing access.
- **ec2_security_group_owner_id** (*string*) – The ID of the owner of the EC2 security from which you are removing access.
- **cidr_ip** (*string*) – The CIDR block from which you are removing access. See http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Return type *bool*

Returns True if successful.

boto.rds.connect_to_region (*region_name*, ****kw_params**)

Given a valid region name, return a *boto.rds.RDSConnection*. Any additional parameters after the *region_name* are passed on to the connect method of the region object.

Type *str*

Parameters **region_name** – The name of the region to connect to.

Return type *boto.rds.RDSConnection* or *None*

Returns A connection to the given region, or *None* if an invalid region name is given

boto.rds.regions ()

Get all available regions for the RDS service.

Return type list

Returns A list of `boto.rds.regioninfo.RDSRegionInfo`

boto.rds.dbinstance

class `boto.rds.dbinstance.DBInstance` (*connection=None, id=None*)

Represents a RDS DBInstance

Properties reference available from the AWS documentation at <http://goo.gl/sC2Kn>

Variables

- **connection** – connection
- **id** – The name and identifier of the DBInstance
- **create_time** – The date and time of creation
- **engine** – The database engine being used
- **status** – The status of the database in a string. e.g. “available”
- **allocated_storage** – The size of the disk in gigabytes (int).
- **auto_minor_version_upgrade** – Indicates that minor version patches are applied automatically.
- **endpoint** – A tuple that describes the hostname and port of the instance. This is only available when the database is in status “available”.
- **instance_class** – Contains the name of the compute and memory capacity class of the DB Instance.
- **master_username** – The username that is set as master username at creation time.
- **parameter_groups** – Provides the list of DB Parameter Groups applied to this DB Instance.
- **security_groups** – Provides List of DB Security Group elements containing only DBSecurityGroup.Name and DBSecurityGroup.Status subelements.
- **availability_zone** – Specifies the name of the Availability Zone the DB Instance is located in.
- **backup_retention_period** – Specifies the number of days for which automatic DB Snapshots are retained.
- **preferred_backup_window** – Specifies the daily time range during which automated backups are created if automated backups are enabled, as determined by the backup_retention_period.
- **preferred_maintenance_window** – Specifies the weekly time range (in UTC) during which system maintenance can occur. (string)
- **latest_restorable_time** – Specifies the latest time to which a database can be restored with point-in-time restore. (string)
- **multi_az** – Boolean that specifies if the DB Instance is a Multi-AZ deployment.
- **iops** – The current number of provisioned IOPS for the DB Instance. Can be None if this is a standard instance.
- **vpc_security_groups** – List of VPC Security Group Membership elements containing only VpcSecurityGroupMembership.VpcSecurityGroupId and VpcSecurityGroupMembership.Status subelements.

- **pending_modified_values** – Specifies that changes to the DB Instance are pending. This element is only included when changes are pending. Specific changes are identified by subelements.
- **read_replica_dbinstance_identifiers** – List of read replicas associated with this DB instance.
- **status_infos** – The status of a Read Replica. If the instance is not a for a read replica, this will be blank.
- **character_set_name** – If present, specifies the name of the character set that this instance is associated with.
- **subnet_group** – Specifies information on the subnet group associated with the DB instance, including the name, description, and subnets in the subnet group.
- **engine_version** – Indicates the database engine version.
- **license_model** – License model information for this DB instance.

endElement (*name, value, connection*)

modify (*param_group=None, security_groups=None, preferred_maintenance_window=None, master_password=None, allocated_storage=None, instance_class=None, backup_retention_period=None, preferred_backup_window=None, multi_az=False, iops=None, vpc_security_groups=None, apply_immediately=False, new_instance_id=None*)
 Modify this DBInstance.

Parameters

- **param_group** (*str*) – Name of DBParameterGroup to associate with this DBInstance.
- **security_groups** (*list of str or list of DBSecurityGroup objects*) – List of names of DBSecurityGroup to authorize on this DBInstance.
- **preferred_maintenance_window** (*str*) – The weekly time range (in UTC) during which maintenance can occur. Default is Sun:05:00-Sun:09:00
- **master_password** (*str*) – Password of master user for the DBInstance. Must be 4-15 alphanumeric characters.
- **allocated_storage** (*int*) – The new allocated storage size, in GBs. Valid values are [5-1024]
- **instance_class** (*str*) – The compute and memory capacity of the DBInstance. Changes will be applied at next maintenance window unless *apply_immediately* is True.
 Valid values are:
 - db.m1.small
 - db.m1.large
 - db.m1.xlarge
 - db.m2.xlarge
 - db.m2.2xlarge
 - db.m2.4xlarge
- **apply_immediately** (*bool*) – If true, the modifications will be applied as soon as possible rather than waiting for the next preferred maintenance window.

- **new_instance_id** (*str*) – The new DB instance identifier.
- **backup_retention_period** (*int*) – The number of days for which automated backups are retained. Setting this to zero disables automated backups.
- **preferred_backup_window** (*str*) – The daily time range during which automated backups are created (if enabled). Must be in h24:mi-hh24:mi format (UTC).
- **multi_az** (*bool*) – If True, specifies the DB Instance will be deployed in multiple availability zones.
- **iops** (*int*) – The amount of IOPS (input/output operations per second) to Provisioned for the DB Instance. Can be modified at a later date.

Must scale linearly. For every 1000 IOPS provision, you must allocated 100 GB of storage space. This scales up to 1 TB / 10 000 IOPS for MySQL and Oracle. MSSQL is limited to 700 GB / 7 000 IOPS.

If you specify a value, it must be at least 1000 IOPS and you must allocate 100 GB of storage.

- **vpc_security_groups** (*list*) – List of VPCSecurityGroupMembership that this DBInstance is a memberof.

Return type *boto.rds.dbinstance.DBInstance*

Returns The modified db instance.

parameter_group

Provide backward compatibility for previous parameter_group attribute.

reboot ()

Reboot this DBInstance

Return type *boto.rds.dbsnapshot.DBSnapshot*

Returns The newly created DBSnapshot

security_group

Provide backward compatibility for previous security_group attribute.

snapshot (*snapshot_id*)

Create a new DB snapshot of this DBInstance.

Parameters **identifier** (*string*) – The identifier for the DBSnapshot

Return type *boto.rds.dbsnapshot.DBSnapshot*

Returns The newly created DBSnapshot

startElement (*name, attrs, connection*)

stop (*skip_final_snapshot=False, final_snapshot_id=""*)

Delete this DBInstance.

Parameters

- **skip_final_snapshot** (*bool*) – This parameter determines whether a final db snapshot is created before the instance is deleted. If True, no snapshot is created. If False, a snapshot is created before deleting the instance.
- **final_snapshot_id** (*str*) – If a final snapshot is requested, this is the identifier used for that snapshot.

Return type *boto.rds.dbinstance.DBInstance*

Returns The deleted db instance.

update (*validate=False*)

Update the DB instance's status information by making a call to fetch the current instance attributes from the service.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the instance the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.

class boto.rds.dbinstance.**PendingModifiedValues**

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.rds.dbinstance.**ReadReplicaDBInstanceIdentifiers**

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.rds.dbsecuritygroup

Represents an DBSecurityGroup

class boto.rds.dbsecuritygroup.**DBSecurityGroup** (*connection=None, owner_id=None, name=None, description=None*)

Represents an RDS database security group

Properties reference available from the AWS documentation at http://docs.amazonwebservices.com/AmazonRDS/latest/APIReference/API_DeleteDBSecurityGroup.html

Variables

- **Status** – The current status of the security group. Possible values are [active, ?]. Reference documentation lacks specifics of possibilities
- **connection** – *boto.rds.RDSConnection* associated with the current object
- **description** – The description of the security group
- **ec2_groups** – List of *EC2 Security Group* objects that this security group PERMITS
- **ip_ranges** – List of *boto.rds.dbsecuritygroup.IPRange* objects (containing CIDR addresses) that this security group PERMITS
- **name** – Name of the security group
- **owner_id** – ID of the owner of the security group. Can be 'None'

authorize (*cidr_ip=None, ec2_group=None*)

Add a new rule to this DBSecurity group. You need to pass in either a CIDR block to authorize or and EC2 SecurityGroup.

Parameters

- **cidr_ip** (*string*) – A valid CIDR IP range to authorize
- **ec2_group** (*boto.ec2.securitygroup.SecurityGroup*) – An EC2 security group to authorize

Return type *bool*

Returns True if successful.

delete ()

endElement (*name, value, connection*)

revoke (*cidr_ip=None, ec2_group=None*)

Revoke access to a CIDR range or EC2 SecurityGroup. You need to pass in either a CIDR block or an EC2 SecurityGroup from which to revoke access.

Parameters

- **cidr_ip** (*string*) – A valid CIDR IP range to revoke
- **ec2_group** (*boto.ec2.securitygroup.SecurityGroup*) – An EC2 security group to revoke

Return type `bool`

Returns True if successful.

startElement (*name, attrs, connection*)

class `boto.rds.dbsecuritygroup.EC2SecurityGroup` (*parent=None*)

Describes an EC2 security group for use in a DBSecurityGroup

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.rds.dbsecuritygroup.IPRange` (*parent=None*)

Describes a CIDR address range for use in a DBSecurityGroup

Variables **cidr_ip** – IP Address range

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.rds.dbsnapshot

class `boto.rds.dbsnapshot.DBSnapshot` (*connection=None, id=None*)

Represents a RDS DB Snapshot

Properties reference available from the AWS documentation at http://docs.amazonwebservices.com/AmazonRDS/latest/APIReference/API_DBSnapshot.html

Variables

- **engine_version** – Specifies the version of the database engine
- **license_model** – License model information for the restored DB instance
- **allocated_storage** – Specifies the allocated storage size in gigabytes (GB)
- **availability_zone** – Specifies the name of the Availability Zone the DB Instance was located in at the time of the DB Snapshot
- **connection** – `boto.rds.RDSConnection` associated with the current object
- **engine** – Specifies the name of the database engine
- **id** – Specifies the identifier for the DB Snapshot (`DBSnapshotIdentifier`)
- **instance_create_time** – Specifies the time (UTC) when the snapshot was taken
- **instance_id** – Specifies the the `DBInstanceIdentifier` of the DB Instance this DB Snapshot was created from (`DBInstanceIdentifier`)
- **master_username** – Provides the master username for the DB Instance

- **port** – Specifies the port that the database engine was listening on at the time of the snapshot
- **snapshot_create_time** – Provides the time (UTC) when the snapshot was taken
- **status** – Specifies the status of this DB Snapshot. Possible values are [available, backing-up, creating, deleted, deleting, failed, modifying, rebooting, resetting-master-credentials]
- **iops** – Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.
- **option_group_name** – Provides the option group name for the DB snapshot.
- **percent_progress** – The percentage of the estimated data that has been transferred.
- **snapshot_type** – Provides the type of the DB snapshot.
- **source_region** – The region that the DB snapshot was created in or copied from.
- **vpc_id** – Provides the Vpc Id associated with the DB snapshot.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

update (*validate=False*)

Update the DB snapshot's status information by making a call to fetch the current snapshot attributes from the service.

Parameters **validate** (*bool*) – By default, if EC2 returns no data about the instance the update method returns quietly. If the validate param is True, however, it will raise a ValueError exception if no data is returned from EC2.

boto.rds.event

class boto.rds.event.**Event** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.rds.parametergroup

class boto.rds.parametergroup.**Parameter** (*group=None, name=None*)

Represents a RDS Parameter

ValidApplyMethods = ['immediate', 'pending-reboot']

ValidApplyTypes = ['static', 'dynamic']

ValidSources = ['user', 'system', 'engine-default']

ValidTypes = {'boolean': <type 'bool'>, 'integer': <type 'int'>, 'string': <type 's

apply (*immediate=False*)

endElement (*name, value, connection*)

get_value ()

merge (*d, i*)

```
set_value (value)
startElement (name, attrs, connection)
value
```

```
class boto.rds.parametergroup.ParameterGroup (connection=None)
```

```
add_param (name, value, apply_method)
endElement (name, value, connection)
get_params ()
modifiable ()
startElement (name, attrs, connection)
```

4.138.7 Redshift

boto.redshift

```
boto.redshift.connect_to_region (region_name, **kw_params)
```

```
boto.redshift.regions ()
```

Get all available regions for the AWS Redshift service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

boto.redshift.layer1

```
class boto.redshift.layer1.RedshiftConnection (**kwargs)
```

Amazon Redshift **Overview** This is an interface reference for Amazon Redshift. It contains documentation for one of the programming or command line interfaces you can use to manage Amazon Redshift clusters. Note that Amazon Redshift is asynchronous, which means that some interfaces may require techniques, such as polling or asynchronous callback handlers, to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a change is applied immediately, on the next instance reboot, or during the next maintenance window. For a summary of the Amazon Redshift cluster management interfaces, go to [‘Using the Amazon Redshift Management Interfaces’](#).

Amazon Redshift manages all the work of setting up, operating, and scaling a data warehouse: provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine. You can focus on using your data to acquire new insights for your business and customers.

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the The [‘Amazon Redshift Getting Started Guide’](#)

If you are a database developer, the [‘Amazon Redshift Database Developer Guide’](#) explains how to design, build, query, and maintain the databases that make up your data warehouse.

```
APIVersion = '2012-12-01'
```

```
DefaultRegionEndpoint = 'redshift.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

ResponseError

alias of `boto.exception.JSONResponseError`

authorize_cluster_security_group_ingress (*cluster_security_group_name*,
cidrip=None,
ec2_security_group_name=None,
ec2_security_group_owner_id=None)

Adds an inbound (ingress) rule to an Amazon Redshift security group. Depending on whether the application accessing your cluster is running on the Internet or an EC2 instance, you can authorize inbound access to either a Classless Interdomain Routing (CIDR) IP address range or an EC2 security group. You can add as many as 20 ingress rules to an Amazon Redshift security group.

For an overview of CIDR blocks, see the Wikipedia article on [‘Classless Inter-Domain Routing’](#).

You must also associate the security group with a cluster so that clients running on these IP addresses or the EC2 instance are authorized to connect to the cluster. For information about managing security groups, go to [‘Working with Security Groups’](#) in the Amazon Redshift Management Guide .

Parameters

- **cluster_security_group_name** (*string*) – The name of the security group to which the ingress rule is added.
- **cidrip** (*string*) – The IP range to be added the Amazon Redshift security group.
- **ec2_security_group_name** (*string*) – The EC2 security group to be added the Amazon Redshift security group.
- **ec2_security_group_owner_id** (*string*) – The AWS account number of the owner of the security group specified by the EC2SecurityGroupName parameter. The AWS Access Key ID is not an acceptable value.

Example: `111122223333`

authorize_snapshot_access (*snapshot_identifier*, *account_with_restore_access*, *snapshot_cluster_identifier=None*)

Authorizes the specified AWS customer account to restore the specified snapshot.

For more information about working with snapshots, go to [‘Amazon Redshift Snapshots’](#) in the Amazon Redshift Management Guide .

Parameters

- **snapshot_identifier** (*string*) – The identifier of the snapshot the account is authorized to restore.
- **snapshot_cluster_identifier** (*string*) – The identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.
- **account_with_restore_access** (*string*) – The identifier of the AWS customer account authorized to restore the specified snapshot.

copy_cluster_snapshot (*source_snapshot_identifier*, *target_snapshot_identifier*,
source_snapshot_cluster_identifier=None)

Copies the specified automated cluster snapshot to a new manual cluster snapshot. The source must be an automated snapshot and it must be in the available state.

When you delete a cluster, Amazon Redshift deletes any automated snapshots of the cluster. Also, when the retention period of the snapshot expires, Amazon Redshift automatically deletes it. If you want to keep an automated snapshot for a longer period, you can make a manual copy of the snapshot. Manual snapshots are retained until you delete them.

For more information about working with snapshots, go to [‘Amazon Redshift Snapshots’](#) in the Amazon Redshift Management Guide .

Parameters `source_snapshot_identifier` (*string*) –
The identifier for the source snapshot.

Constraints:

- **Must be the identifier for a valid automated snapshot whose state is *available*.**

Parameters `source_snapshot_cluster_identifier` (*string*) –

The identifier of the cluster the source snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.

Constraints:

- Must be the identifier for a valid cluster.

Parameters `target_snapshot_identifier` (*string*) –

The identifier given to the new manual snapshot.

Constraints:

- Cannot be null, empty, or blank.
- Must contain from 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for the AWS account that is making the request.

create_cluster (*cluster_identifier*, *node_type*, *master_username*, *master_user_password*, *db_name=None*, *cluster_type=None*, *cluster_security_groups=None*, *vpc_security_group_ids=None*, *cluster_subnet_group_name=None*, *availability_zone=None*, *preferred_maintenance_window=None*, *cluster_parameter_group_name=None*, *automated_snapshot_retention_period=None*, *port=None*, *cluster_version=None*, *allow_version_upgrade=None*, *number_of_nodes=None*, *publicly_accessible=None*, *encrypted=None*, *hsm_client_certificate_identifier=None*, *hsm_configuration_identifier=None*, *elastic_ip=None*)

Creates a new cluster. To create the cluster in virtual private cloud (VPC), you must provide cluster subnet group name. If you don't provide a cluster subnet group name or the cluster security group parameter, Amazon Redshift creates a non-VPC cluster, it associates the default cluster security group with the cluster. For more information about managing clusters, go to '[Amazon Redshift Clusters](#)' in the Amazon Redshift Management Guide .

Parameters `db_name` (*string*) –

The name of the first database to be created when the cluster is created.

To create additional databases after the cluster is created, connect to the cluster with a SQL client and use SQL commands to create a database. For more information, go to '[Create a Database](#)' in the Amazon Redshift Database Developer Guide.

Default: *dev*

Constraints:

- Must contain 1 to 64 alphanumeric characters.
- Must contain only lowercase letters.
- **Cannot be a word that is reserved by the service. A list of reserved words can be found in '[Reserved Words](#)'** in the Amazon Redshift Database Developer Guide.

Parameters `cluster_identifier` (*string*) – A unique identifier for the cluster. You use this identifier to refer to the cluster for any subsequent cluster operations such as deleting or modifying. The identifier also appears in the Amazon Redshift console.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.

- Alphanumeric characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

Example: *myexamplecluster*

Parameters `cluster_type` (*string*) – The type of the cluster. When cluster type is specified as

- *single-node*, the **NumberOfNodes** parameter is not required.
- *multi-node*, the **NumberOfNodes** parameter is required.

Valid Values: *multi-node* | *single-node*

Default: *multi-node*

Parameters `node_type` (*string*) – The node type to be provisioned for the cluster. For information about node types, go to ‘Working with Clusters’_ in the Amazon Redshift Management Guide .

Valid Values: *dw1.xlarge* | *dw1.8xlarge* | *dw2.large* | *dw2.8xlarge*.

Parameters `master_username` (*string*) –

The user name associated with the master user account for the cluster that is being created.

Constraints:

- Must be 1 - 128 alphanumeric characters.
- First character must be a letter.
- **Cannot be a reserved word. A list of reserved words can be found in ‘Reserved Words’_** in the Amazon Redshift Database Developer Guide.

Parameters `master_user_password` (*string*) –

The password associated with the master user account for the cluster that is being created.

Constraints:

- Must be between 8 and 64 characters in length.
- Must contain at least one uppercase letter.
- Must contain at least one lowercase letter.
- Must contain one number.
- **Can be any printable ASCII character (ASCII code 33 to 126) except ‘** (single quote), **”** (double quote), **,** **/**, **@**, or space.

Parameters `cluster_security_groups` (*list*) – A list of security groups to be associated with this cluster.

Default: The default cluster security group for Amazon Redshift.

Parameters `vpc_security_group_ids` (*list*) – A list of Virtual Private Cloud (VPC) security groups to be associated with the cluster.

Default: The default VPC security group is associated with the cluster.

Parameters `cluster_subnet_group_name` (*string*) – The name of a cluster subnet group to be associated with this cluster.

If this parameter is not provided the resulting cluster will be deployed outside virtual private cloud (VPC).

Parameters `availability_zone` (*string*) – The EC2 Availability Zone (AZ) in which you want Amazon Redshift to provision the cluster. For example, if you have several EC2 instances running in a specific Availability Zone, then you might want the cluster to be provisioned in the same zone in order to decrease network latency.

Default: A random, system-chosen Availability Zone in the region that is specified by the endpoint.

Example: *us-east-1d*

Constraint: The specified Availability Zone must be in the same region as the current endpoint.

Parameters `preferred_maintenance_window` (*string*) – The weekly time range (in UTC) during which automated cluster maintenance can occur.

Format: *ddd:hh24:mi-ddd:hh24:mi*

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. The following list shows the time blocks for each region from which the default maintenance windows are assigned.

- **US-East (Northern Virginia) Region:** 03:00-11:00 UTC
- **US-West (Oregon) Region** 06:00-14:00 UTC
- **EU (Ireland) Region** 22:00-06:00 UTC
- **Asia Pacific (Singapore) Region** 14:00-22:00 UTC
- **Asia Pacific (Sydney) Region** 12:00-20:00 UTC
- **Asia Pacific (Tokyo) Region** 17:00-03:00 UTC

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Minimum 30-minute window.

Parameters `cluster_parameter_group_name` (*string*) –

The name of the parameter group to be associated with this cluster.

Default: The default Amazon Redshift cluster parameter group. For information about the default parameter group, go to [‘Working with Amazon Redshift Parameter Groups’](#)

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters `automated_snapshot_retention_period` (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot`.

Default: *1*

Constraints: Must be a value from 0 to 35.

Parameters `port` (*integer*) – The port number on which the cluster accepts incoming connections.

The cluster is accessible only via the JDBC and ODBC connection strings. Part of the connection string requires the port on which the cluster will listen for incoming connections.

Default: *5439*

Valid Values: *1150-65535*

Parameters `cluster_version` (*string*) – The version of the Amazon Redshift engine software that you want to deploy on the cluster.

The version selected runs on all the nodes in the cluster.

Constraints: Only version 1.0 is currently available.

Example: *1.0*

Parameters `allow_version_upgrade` (*boolean*) – If *True*, upgrades can be applied during the maintenance window to the Amazon Redshift engine that is running on the cluster.

When a new version of the Amazon Redshift engine is released, you can request that the service automatically apply upgrades during the maintenance window to the Amazon Redshift engine that is running on your cluster.

Default: *True*

Parameters `number_of_nodes` (*integer*) – The number of compute nodes in the cluster. This parameter is required when the `ClusterType` parameter is specified as *multi-node*.

For information about determining how many nodes you need, go to ‘ Working with Clusters‘_ in the Amazon Redshift Management Guide .

If you don’t specify this parameter, you get a single-node cluster. When requesting a multi-node cluster, you must specify the number of nodes that you want in the cluster.

Default: *1*

Constraints: Value must be at least 1 and no more than 100.

Parameters

- **publicly_accessible** (*boolean*) – If *True*, the cluster can be accessed from a public network.
- **encrypted** (*boolean*) – If *True*, the data in the cluster is encrypted at rest.

Default: *false*

Parameters

- **hsm_client_certificate_identifier** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **hsm_configuration_identifier** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **elastic_ip** (*string*) – The Elastic IP (EIP) address for the cluster.

Constraints: The cluster must be provisioned in EC2-VPC and publicly- accessible through an Internet gateway. For more information about provisioning clusters in EC2-VPC, go to **‘Supported Platforms to Launch Your Cluster‘_** in the Amazon Redshift Management Guide.

create_cluster_parameter_group (*parameter_group_name*, *parameter_group_family*, *description*)

Creates an Amazon Redshift parameter group.

Creating parameter groups is independent of creating clusters. You can associate a cluster with a parameter group when you create the cluster. You can also associate an existing cluster with a parameter group after the cluster is created by using `ModifyCluster`.

Parameters in the parameter group define specific behavior that applies to the databases you create on the cluster. For more information about managing parameter groups, go to **‘Amazon Redshift Parameter Groups‘_** in the Amazon Redshift Management Guide .

Parameters `parameter_group_name` (*string*) –

The name of the cluster parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique within your AWS account.

This value is stored as a lower-case string.

Parameters `parameter_group_family` (*string*) – The Amazon Redshift engine version to which the cluster parameter group applies. The cluster engine version determines the set of parameters.

To get a list of valid parameter group family names, you can call `DescribeClusterParameterGroups`.

By default, Amazon Redshift returns a list of all the parameter groups that are owned by your AWS account, including the default parameter groups for each Amazon Redshift engine version. The parameter group family names associated with the default parameter groups provide you the valid values. For example, a valid family name is “redshift-1.0”.

Parameters `description` (*string*) – A description of the parameter group.

create_cluster_security_group (*cluster_security_group_name, description*)

Creates a new Amazon Redshift security group. You use security groups to control access to non-VPC clusters.

For information about managing security groups, go to ‘[Amazon Redshift Cluster Security Groups](#)’ in the Amazon Redshift Management Guide .

Parameters `cluster_security_group_name` (*string*) – The name for the security group. Amazon Redshift stores the value as a lowercase string.

Constraints:

- Must contain no more than 255 alphanumeric characters or hyphens.
- Must not be “Default”.
- **Must be unique for all security groups that are created by your AWS** account.

Example: *examplesecuritygroup*

Parameters `description` (*string*) – A description for the security group.

create_cluster_snapshot (*snapshot_identifier, cluster_identifier*)

Creates a manual snapshot of the specified cluster. The cluster must be in the *available* state.

For more information about working with snapshots, go to ‘[Amazon Redshift Snapshots](#)’ in the Amazon Redshift Management Guide .

Parameters `snapshot_identifier` (*string*) – A unique identifier for the snapshot that you are requesting. This identifier must be unique for all snapshots within the AWS account.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: *my-snapshot-id*

Parameters `cluster_identifier` (*string*) – The cluster identifier for which you want a snapshot.

create_cluster_subnet_group (*cluster_subnet_group_name, description, subnet_ids*)

Creates a new Amazon Redshift subnet group. You must provide a list of one or more subnets in your existing Amazon Virtual Private Cloud (Amazon VPC) when creating Amazon Redshift subnet group.

For information about subnet groups, go to ‘[Amazon Redshift Cluster Subnet Groups](#)’ in the Amazon Redshift Management Guide .

Parameters `cluster_subnet_group_name` (*string*) – The name for the subnet group. Amazon Redshift stores the value as a lowercase string.

Constraints:

- Must contain no more than 255 alphanumeric characters or hyphens.
- Must not be “Default”.
- **Must be unique for all subnet groups that are created by your AWS** account.

Example: *examplesubnetgroup*

Parameters

- **description** (*string*) – A description for the subnet group.
- **subnet_ids** (*list*) – An array of VPC subnet IDs. A maximum of 20 subnets

can be modified in a single request.

create_event_subscription (*subscription_name*, *sns_topic_arn*, *source_type=None*, *source_ids=None*, *event_categories=None*, *severity=None*, *enabled=None*)

Creates an Amazon Redshift event notification subscription. This action requires an ARN (Amazon Resource Name) of an Amazon SNS topic created by either the Amazon Redshift console, the Amazon SNS console, or the Amazon SNS API. To obtain an ARN with Amazon SNS, you must create a topic in Amazon SNS and subscribe to the topic. The ARN is displayed in the SNS console.

You can specify the source type, and lists of Amazon Redshift source IDs, event categories, and event severities. Notifications will be sent for all events you want that match those criteria. For example, you can specify source type = cluster, source ID = my-cluster-1 and mycluster2, event categories = Availability, Backup, and severity = ERROR. The subscription will only send notifications for those ERROR events in the Availability and Backup categories for the specified clusters.

If you specify both the source type and source IDs, such as source type = cluster and source identifier = my-cluster-1, notifications will be sent for all the cluster events for my-cluster-1. If you specify a source type but do not specify a source identifier, you will receive notice of the events for the objects of that type in your AWS account. If you do not specify either the SourceType nor the SourceIdentifier, you will be notified of events generated from all Amazon Redshift sources belonging to your AWS account. You must specify a source type if you specify a source ID.

Parameters *subscription_name* (*string*) –

The name of the event subscription to be created.

Constraints:

- Cannot be null, empty, or blank.
- Must contain from 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters

- **sns_topic_arn** (*string*) – The Amazon Resource Name (ARN) of the Amazon SNS topic used to transmit the event notifications. The ARN is created by Amazon SNS when you create a topic and subscribe to it.
- **source_type** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a cluster, you would set this parameter to cluster. If this value is not specified, events are returned for all Amazon Redshift objects in your AWS account. You must specify a source type in order to specify source IDs.

Valid values: cluster, cluster-parameter-group, cluster-security-group, and cluster-snapshot.

Parameters *source_ids* (*list*) – A list of one or more identifiers of Amazon Redshift source objects. All of the objects must be of the same type as was specified in the source type parameter. The event subscription will return only events generated by the specified objects. If not specified, then events are returned for all objects within the source type specified.

Example: my-cluster-1, my-cluster-2

Example: my-snapshot-20131010

Parameters *event_categories* (*list*) – Specifies the Amazon Redshift event categories to be published by the event notification subscription.

Values: Configuration, Management, Monitoring, Security

Parameters *severity* (*string*) – Specifies the Amazon Redshift event severity to be published by the event notification subscription.

Values: ERROR, INFO

Parameters **enabled** (*boolean*) – A Boolean value; set to *True* to activate the subscription, set to *False* to create the subscription but not active it.

create_hsm_client_certificate (*hsm_client_certificate_identifier*)

Creates an HSM client certificate that an Amazon Redshift cluster will use to connect to the client's HSM in order to store and retrieve the keys used to encrypt the cluster databases.

The command returns a public key, which you must store in the HSM. In addition to creating the HSM certificate, you must create an Amazon Redshift HSM configuration that provides a cluster the information needed to store and use encryption keys in the HSM. For more information, go to **'Hardware Security Modules'** in the Amazon Redshift Management Guide.

Parameters **hsm_client_certificate_identifier** (*string*) – The identifier to be assigned to the new HSM client certificate that the cluster will use to connect to the HSM to use the database encryption keys.

create_hsm_configuration (*hsm_configuration_identifier*, *description*, *hsm_ip_address*,
hsm_partition_name, *hsm_partition_password*,
hsm_server_public_certificate)

Creates an HSM configuration that contains the information required by an Amazon Redshift cluster to store and use database encryption keys in a Hardware Security Module (HSM). After creating the HSM configuration, you can specify it as a parameter when creating a cluster. The cluster will then store its encryption keys in the HSM.

In addition to creating an HSM configuration, you must also create an HSM client certificate. For more information, go to **'Hardware Security Modules'** in the Amazon Redshift Management Guide.

Parameters

- **hsm_configuration_identifier** (*string*) – The identifier to be assigned to the new Amazon Redshift HSM configuration.
- **description** (*string*) – A text description of the HSM configuration to be created.
- **hsm_ip_address** (*string*) – The IP address that the Amazon Redshift cluster must use to access the HSM.
- **hsm_partition_name** (*string*) – The name of the partition in the HSM where the Amazon Redshift clusters will store their database encryption keys.
- **hsm_partition_password** (*string*) – The password required to access the HSM partition.
- **hsm_server_public_certificate** (*string*) – The HSMs public certificate file. When using Cloud HSM, the file name is `server.pem`.

delete_cluster (*cluster_identifier*, *skip_final_cluster_snapshot=None*, *final_cluster_snapshot_identifier=None*)

Deletes a previously provisioned cluster. A successful response from the web service indicates that the request was received correctly. If a final cluster snapshot is requested the status of the cluster will be "final-snapshot" while the snapshot is being taken, then it's "deleting" once Amazon Redshift begins deleting the cluster. Use `DescribeClusters` to monitor the status of the deletion. The delete operation cannot be canceled or reverted once submitted. For more information about managing clusters, go to **'Amazon Redshift Clusters'** in the Amazon Redshift Management Guide.

Parameters **cluster_identifier** (*string*) – The identifier of the cluster to be deleted.

Constraints:

- Must contain lowercase characters.
- Must contain from 1 to 63 alphanumeric characters or hyphens.
- First character must be a letter.

- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters `skip_final_cluster_snapshot` (*boolean*) – Determines whether a final snapshot of the cluster is created before Amazon Redshift deletes the cluster. If *True*, a final cluster snapshot is not created. If *False*, a final cluster snapshot is created before the cluster is deleted.

Default: *False*

Parameters `final_cluster_snapshot_identifier` (*string*) –

The identifier of the final snapshot that is to be created immediately before deleting the cluster. If this parameter is provided, `SkipFinalClusterSnapshot` must be *False*.

Constraints:

- Must be 1 to 255 alphanumeric characters.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

delete_cluster_parameter_group (*parameter_group_name*)

Deletes a specified Amazon Redshift parameter group.

Parameters `parameter_group_name` (*string*) –

The name of the parameter group to be deleted.

Constraints:

- Must be the name of an existing cluster parameter group.
- Cannot delete a default cluster parameter group.

delete_cluster_security_group (*cluster_security_group_name*)

Deletes an Amazon Redshift security group.

For information about managing security groups, go to [‘Amazon Redshift Cluster Security Groups’](#) in the Amazon Redshift Management Guide .

Parameters `cluster_security_group_name` (*string*) – The name of the cluster security group to be deleted.

delete_cluster_snapshot (*snapshot_identifier*, *snapshot_cluster_identifier=None*)

Deletes the specified manual snapshot. The snapshot must be in the *available* state, with no other users authorized to access the snapshot.

Unlike automated snapshots, manual snapshots are retained even after you delete your cluster. Amazon Redshift does not delete your manual snapshots. You must delete manual snapshot explicitly to avoid getting charged. If other accounts are authorized to access the snapshot, you must revoke all of the authorizations before you can delete the snapshot.

Parameters `snapshot_identifier` (*string*) – The unique identifier of the manual snapshot to be deleted.

Constraints: Must be the name of an existing snapshot that is in the *available* state.

Parameters `snapshot_cluster_identifier` (*string*) – The unique identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than * for the cluster name.

Constraints: Must be the name of valid cluster.

delete_cluster_subnet_group (*cluster_subnet_group_name*)

Deletes the specified cluster subnet group.

Parameters `cluster_subnet_group_name` (*string*) – The name of the cluster subnet group name to be deleted.

delete_event_subscription (*subscription_name*)

Deletes an Amazon Redshift event notification subscription.

Parameters **subscription_name** (*string*) – The name of the Amazon Redshift event notification subscription to be deleted.

delete_hsm_client_certificate (*hsm_client_certificate_identifier*)

Deletes the specified HSM client certificate.

Parameters **hsm_client_certificate_identifier** (*string*) – The identifier of the HSM client certificate to be deleted.

delete_hsm_configuration (*hsm_configuration_identifier*)

Deletes the specified Amazon Redshift HSM configuration.

Parameters **hsm_configuration_identifier** (*string*) – The identifier of the Amazon Redshift HSM configuration to be deleted.

describe_cluster_parameter_groups (*parameter_group_name=None, max_records=None, marker=None*)

Returns a list of Amazon Redshift parameter groups, including parameter groups you created and the default parameter group. For each parameter group, the response includes the parameter group name, description, and parameter group family name. You can optionally specify a name to retrieve the description of a specific parameter group.

For more information about managing parameter groups, go to [‘Amazon Redshift Parameter Groups’](#) in the Amazon Redshift Management Guide .

Parameters

- **parameter_group_name** (*string*) – The name of a specific parameter group for which to return details. By default, details about all parameter groups and the default parameter group are returned.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters **marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeClusterParameterGroups` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_cluster_parameters (*parameter_group_name, source=None, max_records=None, marker=None*)

Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group. For each parameter the response includes information such as parameter name, description, data type, value, whether the parameter value is modifiable, and so on.

You can specify source filter to retrieve parameters of only specific type. For example, to retrieve parameters that were modified by a user action such as from `ModifyClusterParameterGroup`, you can specify source equal to user .

For more information about managing parameter groups, go to [‘Amazon Redshift Parameter Groups’](#) in the Amazon Redshift Management Guide .

Parameters

- **parameter_group_name** (*string*) – The name of a cluster parameter group for which to return details.

- **source** (*string*) – The parameter types to return. Specify *user* to show parameters that are different from the default. Similarly, specify *engine-default* to show parameters that are the same as the default parameter group.

Default: All parameter types returned.

Valid Values: *user* | *engine-default*

Parameters max_records (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusterParameters request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_cluster_security_groups (*cluster_security_group_name=None, max_records=None, marker=None*)

Returns information about Amazon Redshift security groups. If the name of a security group is specified, the response will contain only information about only that security group.

For information about managing security groups, go to [‘Amazon Redshift Cluster Security Groups’](#) in the Amazon Redshift Management Guide .

Parameters cluster_security_group_name (*string*) – The name of a cluster security group for which you are requesting details. You can specify either the **Marker** parameter or a **ClusterSecurityGroupName** parameter, but not both.

Example: *securitygroup1*

Parameters max_records (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusterSecurityGroups request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

Constraints: You can specify either the **ClusterSecurityGroupName** parameter or the **Marker** parameter, but not both.

describe_cluster_snapshots (*cluster_identifier=None, snapshot_identifier=None, snapshot_type=None, start_time=None, end_time=None, max_records=None, marker=None, owner_account=None*)

Returns one or more snapshot objects, which contain metadata about your cluster snapshots. By default, this operation returns information about all snapshots of all clusters that are owned by you AWS customer account. No information is returned for snapshots owned by inactive AWS customer accounts.

Parameters

- **cluster_identifier** (*string*) – The identifier of the cluster for which information about snapshots is requested.

- **snapshot_identifier** (*string*) – The snapshot identifier of the snapshot about which to return information.
- **snapshot_type** (*string*) – The type of snapshots for which you are requesting information. By default, snapshots of all types are returned.

Valid Values: *automated* | *manual*

Parameters start_time (*timestamp*) – A value that requests only snapshots created at or after the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#) _

Example: *2012-07-16T18:00:00Z*

Parameters end_time (*timestamp*) – A time value that requests only snapshots created at or before the specified time. The time value is specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#) _

Example: *2012-07-16T18:00:00Z*

Parameters max_records (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters

- **marker** (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusterSnapshots request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.
- **owner_account** (*string*) – The AWS customer account used to create or copy the snapshot. Use this field to filter the results to snapshots owned by a particular account. To describe snapshots you own, either specify your AWS customer account, or do not specify the parameter.

describe_cluster_subnet_groups (*cluster_subnet_group_name=None, max_records=None, marker=None*)

Returns one or more cluster subnet group objects, which contain metadata about your cluster subnet groups. By default, this operation returns information about all cluster subnet groups that are defined in you AWS account.

Parameters

- **cluster_subnet_group_name** (*string*) – The name of the cluster subnet group for which information is requested.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusterSubnetGroups request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the

returned marker value in the *Marker* parameter and retrying the request.

describe_cluster_versions (*cluster_version=None, cluster_parameter_group_family=None, max_records=None, marker=None*)

Returns descriptions of the available Amazon Redshift cluster versions. You can call this operation even before creating any clusters to learn more about the Amazon Redshift versions. For more information about managing clusters, go to ‘**Amazon Redshift Clusters**’ in the Amazon Redshift Management Guide

Parameters *cluster_version* (*string*) – The specific cluster version to return.

Example: *1.0*

Parameters *cluster_parameter_group_family* (*string*) –

The name of a specific cluster parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters *max_records* (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters *marker* (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusterVersions request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_clusters (*cluster_identifier=None, max_records=None, marker=None*)

Returns properties of provisioned clusters including general cluster properties, cluster database properties, maintenance and backup properties, and security and access properties. This operation supports pagination. For more information about managing clusters, go to ‘**Amazon Redshift Clusters**’ in the Amazon Redshift Management Guide .

Parameters *cluster_identifier* (*string*) – The unique identifier of a cluster whose properties you are requesting. This parameter is case sensitive.

The default is that all clusters defined for an account are returned.

Parameters *max_records* (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters *marker* (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeClusters request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

Constraints: You can specify either the **ClusterIdentifier** parameter or the **Marker** parameter, but not both.

describe_default_cluster_parameters (*parameter_group_family*, *max_records=None*,
marker=None)

Returns a list of parameter settings for the specified parameter group family.

For more information about managing parameter groups, go to [‘Amazon Redshift Parameter Groups’](#) in the Amazon Redshift Management Guide .

Parameters

- **parameter_group_family** (*string*) – The name of the cluster parameter group family.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeDefaultClusterParameters request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_event_categories (*source_type=None*)

Displays a list of event categories for all event source types, or for a specified source type. For a list of the event categories and source types, go to [‘Amazon Redshift Event Notifications’](#).

Parameters source_type (*string*) – The source type, such as cluster or parameter group, to which the described event categories apply.

Valid values: cluster, snapshot, parameter group, and security group.

describe_event_subscriptions (*subscription_name=None*, *max_records=None*,
marker=None)

Lists descriptions of all the Amazon Redshift event notifications subscription for a customer account. If you specify a subscription name, lists the description for that subscription.

Parameters

- **subscription_name** (*string*) – The name of the Amazon Redshift event notification subscription to be described.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeEventSubscriptions request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_events (*source_identifier=None*, *source_type=None*, *start_time=None*, *end_time=None*,
duration=None, *max_records=None*, *marker=None*)

Returns events related to clusters, security groups, snapshots, and parameter groups for the past 14 days. Events specific to a particular cluster, security group, snapshot or parameter group can be obtained by providing the name as a parameter. By default, the past hour of events are returned.

Parameters `source_identifier` (*string*) –

The identifier of the event source for which events will be returned. If this parameter is not specified, then all sources are included in the response.

Constraints:

If `SourceIdentifier` is supplied, `SourceType` must also be provided.

- Specify a cluster identifier when `SourceType` is `cluster`.
- **Specify a cluster security group name when `SourceType` is `'cluster-security-group'`.**
- **Specify a cluster parameter group name when `SourceType` is `'cluster-parameter-group'`.**
- **Specify a cluster snapshot identifier when `SourceType` is `'cluster-snapshot'`.**

Parameters `source_type` (*string*) –

The event source to retrieve events for. If no value is specified, all events are returned.

Constraints:

If `SourceType` is supplied, `SourceIdentifier` must also be provided.

- Specify `cluster` when `SourceIdentifier` is a cluster identifier.
- **Specify `cluster-security-group` when `SourceIdentifier` is a cluster** security group name.
- **Specify `cluster-parameter-group` when `SourceIdentifier` is a cluster** parameter group name.
- **Specify `cluster-snapshot` when `SourceIdentifier` is a cluster** snapshot identifier.

Parameters `start_time` (*timestamp*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#) _

Example: `2009-07-08T18:00Z`

Parameters `end_time` (*timestamp*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#) _

Example: `2009-07-08T18:00Z`

Parameters `duration` (*integer*) – The number of minutes prior to the time of the request for which to retrieve events. For example, if the request is sent at 18:00 and you specify a duration of 60, then only events which have occurred after 17:00 will be returned.

Default: `60`

Parameters `max_records` (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified `MaxRecords` value, a value is returned in a `marker` field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: `100`

Constraints: minimum 20, maximum 100.

Parameters `marker` (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeEvents` request exceed the value specified in `MaxRecords`, AWS returns a value in the `Marker` field of the response. You can retrieve the next set of response records by providing the returned marker value in the `Marker` parameter and retrying the request.

describe_hsm_client_certificates (*hsm_client_certificate_identifier=None, max_records=None, marker=None*)

Returns information about the specified HSM client certificate. If no certificate ID is specified, returns information about all the HSM certificates owned by your AWS customer account.

Parameters

- **hsm_client_certificate_identifier** (*string*) – The identifier of a specific HSM client certificate for which you want information. If no identifier is specified, information is returned for all HSM client certificates owned by your AWS customer account.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeHsmClientCertificates` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_hsm_configurations (*hsm_configuration_identifier=None, max_records=None, marker=None*)

Returns information about the specified Amazon Redshift HSM configuration. If no configuration ID is specified, returns information about all the HSM configurations owned by your AWS customer account.

Parameters

- **hsm_configuration_identifier** (*string*) – The identifier of a specific Amazon Redshift HSM configuration to be described. If no identifier is specified, information is returned for all HSM configurations owned by your AWS customer account.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeHsmConfigurations` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_logging_status (*cluster_identifier*)

Describes whether information, such as queries and connection attempts, is being logged for the specified Amazon Redshift cluster.

Parameters cluster_identifier (*string*) – The identifier of the cluster to get the logging status from.

Example: *examplecluster*

describe_orderable_cluster_options (*cluster_version=None, node_type=None, max_records=None, marker=None*)

Returns a list of orderable cluster options. Before you create a new cluster you can use this operation to find what options are available, such as the EC2 Availability Zones (AZ) in the specific AWS region that you can specify, and the node types you can request. The node types differ by available storage, memory, CPU and price. With the cost involved you might want to obtain a list of cluster options in the specific region and specify values when creating a cluster. For more information about managing clusters, go to

‘Amazon Redshift Clusters’ in the Amazon Redshift Management Guide

Parameters `cluster_version` (*string*) – The version filter value. Specify this parameter to show only the available offerings matching the specified version.

Default: All versions.

Constraints: Must be one of the version returned from `DescribeClusterVersions`.

Parameters

- **node_type** (*string*) – The node type filter value. Specify this parameter to show only the available offerings matching the specified node type.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

Parameters `marker` (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeOrderableClusterOptions` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_reserved_node_offerings (*reserved_node_offering_id=None, max_records=None, marker=None*)

Returns a list of the available reserved node offerings by Amazon Redshift with their descriptions including the node type, the fixed and recurring costs of reserving the node and duration the node will be reserved for you. These descriptions help you determine which reserve node offering you want to purchase. You then use the unique offering ID in you call to `PurchaseReservedNodeOffering` to reserve one or more nodes for your Amazon Redshift cluster.

For more information about managing parameter groups, go to **‘Purchasing Reserved Nodes’** in the Amazon Redshift Management Guide .

Parameters

- **reserved_node_offering_id** (*string*) – The unique identifier for the offering.
- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: 100

Constraints: minimum 20, maximum 100.

Parameters `marker` (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a `DescribeReservedNodeOfferings` request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_reserved_nodes (*reserved_node_id=None, max_records=None, marker=None*)

Returns the descriptions of the reserved nodes.

Parameters

- **reserved_node_id** (*string*) – Identifier for the node reservation.

- **max_records** (*integer*) – The maximum number of response records to return in each call. If the number of remaining response records exceeds the specified *MaxRecords* value, a value is returned in a *marker* field of the response. You can retrieve the next set of records by retrying the command with the returned marker value.

Default: *100*

Constraints: minimum 20, maximum 100.

Parameters marker (*string*) – An optional parameter that specifies the starting point to return a set of response records. When the results of a DescribeReservedNodes request exceed the value specified in *MaxRecords*, AWS returns a value in the *Marker* field of the response. You can retrieve the next set of response records by providing the returned marker value in the *Marker* parameter and retrying the request.

describe_resize (*cluster_identifier*)

Returns information about the last resize operation for the specified cluster. If no resize operation has ever been initiated for the specified cluster, a *HTTP 404* error is returned. If a resize operation was initiated and completed, the status of the resize remains as *SUCCEEDED* until the next resize.

A resize operation can be requested using ModifyCluster and specifying a different number or type of nodes for the cluster.

Parameters cluster_identifier (*string*) – The unique identifier of a cluster whose resize progress you are requesting. This parameter isn't case-sensitive.

By default, resize operations for all clusters defined for an AWS account are returned.

disable_logging (*cluster_identifier*)

Stops logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster.

Parameters cluster_identifier (*string*) – The identifier of the cluster on which logging is to be stopped.

Example: *examplecluster*

disable_snapshot_copy (*cluster_identifier*)

Disables the automatic copying of snapshots from one region to another region for a specified cluster.

Parameters cluster_identifier (*string*) – The unique identifier of the source cluster that you want to disable copying of snapshots to a destination region.

Constraints: Must be the valid name of an existing cluster that has cross-region snapshot copy enabled.

enable_logging (*cluster_identifier, bucket_name, s3_key_prefix=None*)

Starts logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster.

Parameters cluster_identifier (*string*) – The identifier of the cluster on which logging is to be started.

Example: *examplecluster*

Parameters bucket_name (*string*) –

The name of an existing S3 bucket where the log files are to be stored.

Constraints:

- Must be in the same region as the cluster
- The cluster must have read bucket and put object permissions

Parameters s3_key_prefix (*string*) –

The prefix applied to the log file names.

Constraints:

- Cannot exceed 512 characters

- **Cannot contain spaces(), double quotes (“), single quotes (’), a backslash (), or control characters.** The hexadecimal codes for invalid characters are:
 - x00 to x20
 - x22
 - x27
 - x5c
 - x7f or larger

enable_snapshot_copy (*cluster_identifier*, *destination_region*, *retention_period=None*)

Enables the automatic copy of snapshots from one region to another region for a specified cluster.

Parameters **cluster_identifier** (*string*) – The unique identifier of the source cluster to copy snapshots from.

Constraints: Must be the valid name of an existing cluster that does not already have cross-region snapshot copy enabled.

Parameters **destination_region** (*string*) – The destination region that you want to copy snapshots to.

Constraints: Must be the name of a valid region. For more information, see **‘Regions and End-points’** in the Amazon Web Services General Reference.

Parameters **retention_period** (*integer*) – The number of days to retain automated snapshots in the destination region after they are copied from the source region.

Default: 7.

Constraints: Must be at least 1 and no more than 35.

modify_cluster (*cluster_identifier*, *cluster_type=None*, *node_type=None*, *number_of_nodes=None*, *cluster_security_groups=None*, *vpc_security_group_ids=None*, *master_user_password=None*, *cluster_parameter_group_name=None*, *automated_snapshot_retention_period=None*, *preferred_maintenance_window=None*, *cluster_version=None*, *allow_version_upgrade=None*, *hsm_client_certificate_identifier=None*, *hsm_configuration_identifier=None*, *new_cluster_identifier=None*)

Modifies the settings for a cluster. For example, you can add another security or parameter group, update the preferred maintenance window, or change the master user password. Resetting a cluster password or modifying the security groups associated with a cluster do not need a reboot. However, modifying a parameter group requires a reboot for parameters to take effect. For more information about managing clusters, go to **‘Amazon Redshift Clusters’** in the Amazon Redshift Management Guide

You can also change node type and the number of nodes to scale up or down the cluster. When resizing a cluster, you must specify both the number of nodes and the node type even if one of the parameters does not change. If you specify the same number of nodes and node type that are already configured for the cluster, an error is returned.

Parameters **cluster_identifier** (*string*) – The unique identifier of the cluster to be modified.

Example: *examplecluster*

Parameters **cluster_type** (*string*) – The new cluster type.

When you submit your cluster resize request, your existing cluster goes into a read-only mode. After Amazon Redshift provisions a new cluster based on your resize requirements, there will be outage for a period while the old cluster is deleted and your connection is switched to the new cluster. You can use DescribeResize to track the progress of the resize request.

Valid Values: ‘ multi-node | single-node ‘

Parameters `node_type` (*string*) – The new node type of the cluster. If you specify a new node type, you must also specify the number of nodes parameter also.

When you submit your request to resize a cluster, Amazon Redshift sets access permissions for the cluster to read-only. After Amazon Redshift provisions a new cluster according to your resize requirements, there will be a temporary outage while the old cluster is deleted and your connection is switched to the new cluster. When the new connection is complete, the original access permissions for the cluster are restored. You can use the DescribeResize to track the progress of the resize request.

Valid Values: ‘ dw1.xlarge ‘ | *dw1.8xlarge* | *dw2.large* | *dw2.8xlarge*.

Parameters `number_of_nodes` (*integer*) – The new number of nodes of the cluster. If you specify a new number of nodes, you must also specify the node type parameter also.

When you submit your request to resize a cluster, Amazon Redshift sets access permissions for the cluster to read-only. After Amazon Redshift provisions a new cluster according to your resize requirements, there will be a temporary outage while the old cluster is deleted and your connection is switched to the new cluster. When the new connection is complete, the original access permissions for the cluster are restored. You can use DescribeResize to track the progress of the resize request.

Valid Values: Integer greater than 0.

Parameters `cluster_security_groups` (*list*) –

A list of cluster security groups to be authorized on this cluster. This change is asynchronously applied as soon as possible.

Security groups currently associated with the cluster, and not in the list of groups to apply, will be revoked from the cluster.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters

- `vpc_security_group_ids` (*list*) – A list of virtual private cloud (VPC) security groups to be associated with the cluster.
- `master_user_password` (*string*) –

The new password for the cluster master user. This change is asynchronously applied as soon as possible. Between the time of the request and the completion of the request, the `MasterUserPassword` element exists in the `PendingModifiedValues` element of the operation response.

Default: Uses existing setting.

Constraints:

- Must be between 8 and 64 characters in length.
- Must contain at least one uppercase letter.
- Must contain at least one lowercase letter.
- Must contain one number.
- **Can be any printable ASCII character (ASCII code 33 to 126) except** ‘ (single quote), ” (double quote), , /, @, or space.

Parameters `cluster_parameter_group_name` (*string*) – The name of the cluster parameter group to apply to this cluster. This change is applied only after the cluster is rebooted. To reboot a cluster use RebootCluster.

Default: Uses existing setting.

Constraints: The cluster parameter group must be in the same parameter group family that matches the cluster version.

Parameters `automated_snapshot_retention_period` (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot`.

If you decrease the automated snapshot retention period from its current value, existing automated snapshots that fall outside of the new retention period will be immediately deleted.

Default: Uses existing setting.

Constraints: Must be a value from 0 to 35.

Parameters `preferred_maintenance_window` (*string*) – The weekly time range (in UTC) during which system maintenance can occur, if necessary. If system maintenance is necessary during the window, it may result in an outage.

This maintenance window change is made immediately. If the new maintenance window indicates the current time, there must be at least 120 minutes between the current time and end of the window in order to ensure that pending changes are applied.

Default: Uses existing setting.

Format: `ddd:hh24:mi-ddd:hh24:mi`, for example `wed:07:30-wed:08:00`.

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Must be at least 30 minutes.

Parameters `cluster_version` (*string*) – The new version number of the Amazon Redshift engine to upgrade to.

For major version upgrades, if a non-default cluster parameter group is currently in use, a new cluster parameter group in the cluster parameter group family for the new version must be specified. The new cluster parameter group can be the default for that cluster parameter group family. For more information about managing parameter groups, go to **‘Amazon Redshift Parameter Groups’** in the Amazon Redshift Management Guide .

Example: `1.0`

Parameters `allow_version_upgrade` (*boolean*) – If `True`, upgrades will be applied automatically to the cluster during the maintenance window.

Default: `False`

Parameters

- **`hsm_client_certificate_identifier`** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **`hsm_configuration_identifier`** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **`new_cluster_identifier`** (*string*) – The new identifier for the cluster.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphanumeric characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

Example: `examplecluster`

modify_cluster_parameter_group (*parameter_group_name*, *parameters*)

Modifies the parameters of a parameter group.

For more information about managing parameter groups, go to [‘Amazon Redshift Parameter Groups’](#) in the Amazon Redshift Management Guide .

Parameters

- **parameter_group_name** (*string*) – The name of the parameter group to be modified.
- **parameters** (*list*) – An array of parameters to be modified. A maximum of 20 parameters can be modified in a single request.

For each parameter to be modified, you must supply at least the parameter name and parameter value; other name-value pairs of the parameter are optional.

For the workload management (WLM) configuration, you must supply all the name-value pairs in the `wlm_json_configuration` parameter.

modify_cluster_subnet_group (*cluster_subnet_group_name*, *subnet_ids*, *description=None*)

Modifies a cluster subnet group to include the specified list of VPC subnets. The operation replaces the existing list of subnets with the new list of subnets.

Parameters

- **cluster_subnet_group_name** (*string*) – The name of the subnet group to be modified.
- **description** (*string*) – A text description of the subnet group to be modified.
- **subnet_ids** (*list*) – An array of VPC subnet IDs. A maximum of 20 subnets can be modified in a single request.

modify_event_subscription (*subscription_name*, *sns_topic_arn=None*, *source_type=None*, *source_ids=None*, *event_categories=None*, *severity=None*, *enabled=None*)

Modifies an existing Amazon Redshift event notification subscription.

Parameters

- **subscription_name** (*string*) – The name of the modified Amazon Redshift event notification subscription.
- **sns_topic_arn** (*string*) – The Amazon Resource Name (ARN) of the SNS topic to be used by the event notification subscription.
- **source_type** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a cluster, you would set this parameter to `cluster`. If this value is not specified, events are returned for all Amazon Redshift objects in your AWS account. You must specify a source type in order to specify source IDs.

Valid values: `cluster`, `cluster-parameter-group`, `cluster-security-group`, and `cluster-snapshot`.

Parameters **source_ids** (*list*) – A list of one or more identifiers of Amazon Redshift source objects. All of the objects must be of the same type as was specified in the source type parameter. The event subscription will return only events generated by the specified objects. If not specified, then events are returned for all objects within the source type specified.

Example: `my-cluster-1`, `my-cluster-2`

Example: `my-snapshot-20131010`

Parameters `event_categories` (*list*) – Specifies the Amazon Redshift event categories to be published by the event notification subscription.

Values: Configuration, Management, Monitoring, Security

Parameters `severity` (*string*) – Specifies the Amazon Redshift event severity to be published by the event notification subscription.

Values: ERROR, INFO

Parameters `enabled` (*boolean*) – A Boolean value indicating if the subscription is enabled. *True* indicates the subscription is enabled

modify_snapshot_copy_retention_period (*cluster_identifier, retention_period*)

Modifies the number of days to retain automated snapshots in the destination region after they are copied from the source region.

Parameters `cluster_identifier` (*string*) – The unique identifier of the cluster for which you want to change the retention period for automated snapshots that are copied to a destination region.

Constraints: Must be the valid name of an existing cluster that has cross-region snapshot copy enabled.

Parameters `retention_period` (*integer*) – The number of days to retain automated snapshots in the destination region after they are copied from the source region.

If you decrease the retention period for automated snapshots that are copied to a destination region, Amazon Redshift will delete any existing automated snapshots that were copied to the destination region and that fall outside of the new retention period.

Constraints: Must be at least 1 and no more than 35.

purchase_reserved_node_offering (*reserved_node_offering_id, node_count=None*)

Allows you to purchase reserved nodes. Amazon Redshift offers a predefined set of reserved node offerings. You can purchase one of the offerings. You can call the DescribeReservedNodeOfferings API to obtain the available reserved node offerings. You can call this API by providing a specific reserved node offering and the number of nodes you want to reserve.

For more information about managing parameter groups, go to **‘Purchasing Reserved Nodes’** in the Amazon Redshift Management Guide .

Parameters

- **reserved_node_offering_id** (*string*) – The unique identifier of the reserved node offering you want to purchase.
- **node_count** (*integer*) – The number of reserved nodes you want to purchase.

Default: 1

reboot_cluster (*cluster_identifier*)

Reboots a cluster. This action is taken as soon as possible. It results in a momentary outage to the cluster, during which the cluster status is set to *rebooting*. A cluster event is created when the reboot is completed. Any pending cluster modifications (see ModifyCluster) are applied at this reboot. For more information about managing clusters, go to **‘Amazon Redshift Clusters’** in the Amazon Redshift Management Guide

Parameters `cluster_identifier` (*string*) – The cluster identifier.

reset_cluster_parameter_group (*parameter_group_name, reset_all_parameters=None, parameters=None*)

Sets one or more parameters of the specified parameter group to their default values and sets the source values of the parameters to “engine-default”. To reset the entire parameter group specify the ResetAllParameters parameter. For parameter changes to take effect you must reboot any associated clusters.

Parameters

- **parameter_group_name** (*string*) – The name of the cluster parameter group to be reset.
- **reset_all_parameters** (*boolean*) – If *True*, all parameters in the specified parameter group will be reset to their default values.

Default: *True*

Parameters parameters (*list*) – An array of names of parameters to be reset. If `ResetAllParameters` option is not used, then at least one parameter name must be supplied.

Constraints: A maximum of 20 parameters can be reset in a single request.

restore_from_cluster_snapshot (*cluster_identifier*, *snapshot_identifier*, *snapshot_cluster_identifier=None*, *port=None*, *availability_zone=None*, *allow_version_upgrade=None*, *cluster_subnet_group_name=None*, *publicly_accessible=None*, *owner_account=None*, *hsm_client_certificate_identifier=None*, *hsm_configuration_identifier=None*, *elastic_ip=None*, *cluster_parameter_group_name=None*, *cluster_security_groups=None*, *vpc_security_group_ids=None*, *preferred_maintenance_window=None*, *automated_snapshot_retention_period=None*)

Creates a new cluster from a snapshot. Amazon Redshift creates the resulting cluster with the same configuration as the original cluster from which the snapshot was created, except that the new cluster is created with the default cluster security and parameter group. After Amazon Redshift creates the cluster you can use the `ModifyCluster` API to associate a different security group and different parameter group with the restored cluster.

If you restore a cluster into a VPC, you must provide a cluster subnet group where you want the cluster restored.

For more information about working with snapshots, go to [‘Amazon Redshift Snapshots’](#) in the Amazon Redshift Management Guide .

Parameters cluster_identifier (*string*) – The identifier of the cluster that will be created from restoring the snapshot.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens.
- Alphabetic characters must be lowercase.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.
- Must be unique for all clusters within an AWS account.

Parameters snapshot_identifier (*string*) – The name of the snapshot from which to create the new cluster. This parameter isn’t case sensitive.

Example: *my-snapshot-id*

Parameters

- **snapshot_cluster_identifier** (*string*) – The name of the cluster the source snapshot was created from. This parameter is required if your IAM user has a policy containing a snapshot resource element that specifies anything other than `*` for the cluster name.
- **port** (*integer*) – The port number on which the cluster accepts connections.

Default: The same port as the original cluster.

Constraints: Must be between *1115* and *65535*.

Parameters availability_zone (*string*) – The Amazon EC2 Availability Zone in which to restore the cluster.

Default: A random, system-chosen Availability Zone.

Example: *us-east-1a*

Parameters `allow_version_upgrade` (*boolean*) – If *True*, upgrades can be applied during the maintenance window to the Amazon Redshift engine that is running on the cluster.

Default: *True*

Parameters `cluster_subnet_group_name` (*string*) – The name of the subnet group where you want to cluster restored.

A snapshot of cluster in VPC can be restored only in VPC. Therefore, you must provide subnet group name where you want the cluster restored.

Parameters

- **publicly_accessible** (*boolean*) – If *True*, the cluster can be accessed from a public network.
- **owner_account** (*string*) – The AWS customer account used to create or copy the snapshot. Required if you are restoring a snapshot you do not own, optional if you own the snapshot.
- **hsm_client_certificate_identifier** (*string*) – Specifies the name of the HSM client certificate the Amazon Redshift cluster uses to retrieve the data encryption keys stored in an HSM.
- **hsm_configuration_identifier** (*string*) – Specifies the name of the HSM configuration that contains the information the Amazon Redshift cluster can use to retrieve and store keys in an HSM.
- **elastic_ip** (*string*) – The elastic IP (EIP) address for the cluster.
- **cluster_parameter_group_name** (*string*) –

The name of the parameter group to be associated with this cluster.

Default: The default Amazon Redshift cluster parameter group. For information about the default parameter group, go to **‘Working with Amazon Redshift Parameter Groups’**.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens.
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters `cluster_security_groups` (*list*) – A list of security groups to be associated with this cluster.

Default: The default cluster security group for Amazon Redshift.

Cluster security groups only apply to clusters outside of VPCs.

Parameters `vpc_security_group_ids` (*list*) – A list of Virtual Private Cloud (VPC) security groups to be associated with the cluster.

Default: The default VPC security group is associated with the cluster.

VPC security groups only apply to clusters in VPCs.

Parameters `preferred_maintenance_window` (*string*) – The weekly time range (in UTC) during which automated cluster maintenance can occur.

Format: *ddd:hh24:mi-ddd:hh24:mi*

Default: The value selected for the cluster from which the snapshot was taken. The following list shows the time blocks for each region from which the default maintenance windows are assigned.

- **US-East (Northern Virginia) Region:** 03:00-11:00 UTC
- **US-West (Oregon) Region:** 06:00-14:00 UTC

- **EU (Ireland) Region** 22:00-06:00 UTC
- **Asia Pacific (Singapore) Region** 14:00-22:00 UTC
- **Asia Pacific (Sydney) Region** 12:00-20:00 UTC
- **Asia Pacific (Tokyo) Region** 17:00-03:00 UTC

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Minimum 30-minute window.

Parameters `automated_snapshot_retention_period` (*integer*) – The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with `CreateClusterSnapshot`.

Default: The value selected for the cluster from which the snapshot was taken.

Constraints: Must be a value from 0 to 35.

revoke_cluster_security_group_ingress (*cluster_security_group_name*, *cidrip=None*, *ec2_security_group_name=None*, *ec2_security_group_owner_id=None*)

Revokes an ingress rule in an Amazon Redshift security group for a previously authorized IP range or Amazon EC2 security group. To add an ingress rule, see `AuthorizeClusterSecurityGroupIngress`. For information about managing security groups, go to ‘**Amazon Redshift Cluster Security Groups**’ in the Amazon Redshift Management Guide .

Parameters

- **cluster_security_group_name** (*string*) – The name of the security Group from which to revoke the ingress rule.
- **cidrip** (*string*) – The IP range for which to revoke access. This range must be a valid Classless Inter-Domain Routing (CIDR) block of IP addresses. If *CIDRIP* is specified, *EC2SecurityGroupName* and *EC2SecurityGroupOwnerId* cannot be provided.
- **ec2_security_group_name** (*string*) – The name of the EC2 Security Group whose access is to be revoked. If *EC2SecurityGroupName* is specified, *EC2SecurityGroupOwnerId* must also be provided and *CIDRIP* cannot be provided.
- **ec2_security_group_owner_id** (*string*) – The AWS account number of the owner of the security group specified in the *EC2SecurityGroupName* parameter. The AWS access key ID is not an acceptable value. If *EC2SecurityGroupOwnerId* is specified, *EC2SecurityGroupName* must also be provided. and *CIDRIP* cannot be provided.

Example: `111122223333`

revoke_snapshot_access (*snapshot_identifier*, *account_with_restore_access*, *snapshot_cluster_identifier=None*)

Removes the ability of the specified AWS customer account to restore the specified snapshot. If the account is currently restoring the snapshot, the restore will run to completion.

For more information about working with snapshots, go to ‘**Amazon Redshift Snapshots**’ in the Amazon Redshift Management Guide .

Parameters

- **snapshot_identifier** (*string*) – The identifier of the snapshot that the account can no longer access.
- **snapshot_cluster_identifier** (*string*) – The identifier of the cluster the snapshot was created from. This parameter is required if your IAM user has a

policy containing a snapshot resource element that specifies anything other than * for the cluster name.

- **account_with_restore_access** (*string*) – The identifier of the AWS customer account that can no longer restore the specified snapshot.

rotate_encryption_key (*cluster_identifier*)

Rotates the encryption keys for a cluster.

Parameters **cluster_identifier** (*string*) – The unique identifier of the cluster that you want to rotate the encryption keys for.

Constraints: Must be the name of valid cluster that has encryption enabled.

boto.redshift.exceptions

exception boto.redshift.exceptions.**AccessToSnapshotDenied** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AccessToSnapshotDeniedFault** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationAlreadyExists** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationAlreadyExistsFault** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationNotFound** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationNotFoundFault** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationQuotaExceeded** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**AuthorizationQuotaExceededFault** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**BucketNotFound** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**ClusterAlreadyExists** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**ClusterAlreadyExistsFault** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**ClusterNotFound** (*status*, *reason*, *body=None*, *args)

exception boto.redshift.exceptions.**ClusterNotFoundFault** (*status*, *reason*, *body=None*, *args)

```
exception boto.redshift.exceptions.ClusterParameterGroupAlreadyExists (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterParameterGroupAlreadyExistsFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterParameterGroupNotFound (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterParameterGroupNotFoundFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterParameterGroupQuotaExceeded (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterParameterGroupQuotaExceededFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterQuotaExceeded (status,          reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.ClusterQuotaExceededFault (status,          reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSecurityGroupAlreadyExists (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSecurityGroupAlreadyExistsFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSecurityGroupNotFound (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSecurityGroupNotFoundFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.redshift.exceptions.ClusterSecurityGroupQuotaExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSecurityGroupQuotaExceededFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSnapshotAlreadyExists (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSnapshotAlreadyExistsFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSnapshotNotFound (status,      reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.ClusterSnapshotNotFoundFault (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSnapshotQuotaExceeded (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSnapshotQuotaExceededFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetGroupAlreadyExists (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetGroupAlreadyExistsFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetGroupNotFound (status,      reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetGroupNotFoundFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetGroupQuotaExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.redshift.exceptions.ClusterSubnetGroupQuotaExceededFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetQuotaExceeded (status,   reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ClusterSubnetQuotaExceededFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.CopyToRegionDisabled (status,           reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.EventSubscriptionQuotaExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.HsmClientCertificateAlreadyExists (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.HsmClientCertificateNotFound (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.HsmClientCertificateQuotaExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.HsmConfigurationAlreadyExists (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.HsmConfigurationNotFound (status,   reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.HsmConfigurationQuotaExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.IncompatibleOrderableOptions (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InsufficientClusterCapacity (status,   reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InsufficientClusterCapacityFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.redshift.exceptions.InsufficientS3BucketPolicy (status, reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterParameterGroupState (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterParameterGroupStateFault (status,  
                                                    rea-  
                                                    son,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSecurityGroupState (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSecurityGroupStateFault (status,  
                                                    rea-  
                                                    son,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSnapshotState (status, reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSnapshotStateFault (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterState (status, reason, body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterStateFault (status, reason,  
                                                    body=None, *args)  
exception boto.redshift.exceptions.InvalidClusterSubnetGroupState (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSubnetGroupStateFault (status,  
                                                    rea-  
                                                    son,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSubnetState (status, reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidClusterSubnetStateFault (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.redshift.exceptions.InvalidElasticIp (status, reason, body=None,  
                                                    *args)
```

```
exception boto.redshift.exceptions.InvalidHsmClientCertificateState (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidHsmConfigurationState (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidParameterCombinationFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidRestore (status, reason, body=None, *args)  
exception boto.redshift.exceptions.InvalidRestoreFault (status, reason, body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidS3BucketName (status, reason, body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidS3KeyPrefix (status, reason, body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.InvalidSubnet (status, reason, body=None, *args)  
exception boto.redshift.exceptions.InvalidSubscriptionState (status,      reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.InvalidVPCNetworkState (status,      reason,  
                                                                    body=None, *args)  
exception boto.redshift.exceptions.InvalidVPCNetworkStateFault (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.NumberOfNodesPerClusterLimitExceeded (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.NumberOfNodesPerClusterLimitExceededFault (status,  
                                                                    rea-  
                                                                    son,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.NumberOfNodesQuotaExceeded (status, reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.NumberOfNodesQuotaExceededFault (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)  
exception boto.redshift.exceptions.ReservedNodeAlreadyExists (status,      reason,  
                                                                    body=None,  
                                                                    *args)
```

exception boto.redshift.exceptions.**ReservedNodeAlreadyExistsFault** (*status*,
reason,
body=None,
**args*)

exception boto.redshift.exceptions.**ReservedNodeNotFound** (*status*, *reason*,
body=None, **args*)

exception boto.redshift.exceptions.**ReservedNodeNotFoundFault** (*status*, *reason*,
body=None,
**args*)

exception boto.redshift.exceptions.**ReservedNodeOfferingNotFound** (*status*, *reason*,
body=None,
**args*)

exception boto.redshift.exceptions.**ReservedNodeOfferingNotFoundFault** (*status*,
reason,
body=None,
**args*)

exception boto.redshift.exceptions.**ReservedNodeQuotaExceeded** (*status*, *reason*,
body=None,
**args*)

exception boto.redshift.exceptions.**ReservedNodeQuotaExceededFault** (*status*,
reason,
body=None,
**args*)

exception boto.redshift.exceptions.**ResizeNotFound** (*status*, *reason*, *body=None*, **args*)

exception boto.redshift.exceptions.**ResizeNotFoundFault** (*status*, *reason*, *body=None*,
**args*)

exception boto.redshift.exceptions.**SNSInvalidTopic** (*status*, *reason*, *body=None*,
**args*)

exception boto.redshift.exceptions.**SNSNoAuthorization** (*status*, *reason*, *body=None*,
**args*)

exception boto.redshift.exceptions.**SNSTopicArnNotFound** (*status*, *reason*, *body=None*,
**args*)

exception boto.redshift.exceptions.**SnapshotCopyAlreadyDisabled** (*status*, *reason*,
body=None,
**args*)

exception boto.redshift.exceptions.**SnapshotCopyAlreadyEnabled** (*status*, *reason*,
body=None,
**args*)

exception boto.redshift.exceptions.**SnapshotCopyDisabled** (*status*, *reason*,
body=None, **args*)

exception boto.redshift.exceptions.**SourceNotFound** (*status*, *reason*, *body=None*, **args*)

exception boto.redshift.exceptions.**SubnetAlreadyInUse** (*status*, *reason*, *body=None*,
**args*)

exception boto.redshift.exceptions.**SubscriptionAlreadyExist** (*status*, *reason*,
body=None, **args*)

exception boto.redshift.exceptions.**SubscriptionCategoryNotFound** (*status*, *reason*,
body=None,
**args*)

```
exception boto.redshift.exceptions.SubscriptionEventIdNotFound (status, reason,  
                                                             body=None,  
                                                             *args)  
exception boto.redshift.exceptions.SubscriptionNotFound (status, reason,  
                                                             body=None, *args)  
exception boto.redshift.exceptions.SubscriptionSeverityNotFound (status, reason,  
                                                             body=None,  
                                                             *args)  
exception boto.redshift.exceptions.UnauthorizedOperation (status, reason,  
                                                             body=None, *args)  
exception boto.redshift.exceptions.UnauthorizedOperationFault (status, reason,  
                                                             body=None,  
                                                             *args)  
exception boto.redshift.exceptions.UnknownSnapshotCopyRegion (status, reason,  
                                                             body=None,  
                                                             *args)  
exception boto.redshift.exceptions.UnsupportedOption (status, reason, body=None,  
                                                             *args)  
exception boto.redshift.exceptions.UnsupportedOptionFault (status, reason,  
                                                             body=None, *args)
```

4.138.8 route53

boto.route53.connection

```
class boto.route53.connection.Route53Connection (aws_access_key_id=None,  
                                                  aws_secret_access_key=None,  
                                                  port=None, proxy=None,  
                                                  proxy_port=None,  
                                                  host='route53.amazonaws.com',  
                                                  debug=0, security_token=None,  
                                                  validate_certs=True,  
                                                  https_connection_factory=None,  
                                                  profile_name=None)
```

```
DefaultHost = 'route53.amazonaws.com'
```

The default Route53 API endpoint to connect to.

```
POSTHCXMLBody = '<CreateHealthCheckRequest xmlns="% (xmlns)s">\n <CallerReference>%(cal
```

```
Version = '2013-04-01'
```

Route53 API version.

```
XMLNameSpace = 'https://route53.amazonaws.com/doc/2013-04-01/'
```

XML schema for this Route53 API version.

```
change_rrsets (hosted_zone_id, xml_body)
```

Create or change the authoritative DNS information for this Hosted Zone. Returns a Python data structure with information about the set of changes, including the Change ID.

Parameters

- **hosted_zone_id** (*str*) – The unique identifier for the Hosted Zone

- **xml_body** (*str*) – The list of changes to be made, defined in the XML schema defined by the Route53 service.

create_health_check (*health_check*, *caller_ref=None*)

Create a new Health Check

Parameters

- **health_check** (*HealthCheck*) – HealthCheck object
- **caller_ref** (*str*) – A unique string that identifies the request and that allows failed CreateHealthCheckRequest requests to be retried without the risk of executing the operation twice. If you don't provide a value for this, boto will generate a Type 4 UUID and use that.

create_hosted_zone (*domain_name*, *caller_ref=None*, *comment=""*, *private_zone=False*, *vpc_id=None*, *vpc_region=None*)

Create a new Hosted Zone. Returns a Python data structure with information about the newly created Hosted Zone.

Parameters

- **domain_name** (*str*) – The name of the domain. This should be a fully-specified domain, and should end with a final period as the last label indication. If you omit the final period, Amazon Route 53 assumes the domain is relative to the root. This is the name you have registered with your DNS registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request. A list of strings with the image IDs wanted.
- **caller_ref** (*str*) – A unique string that identifies the request and that allows failed CreateHostedZone requests to be retried without the risk of executing the operation twice. If you don't provide a value for this, boto will generate a Type 4 UUID and use that.
- **comment** (*str*) – Any comments you want to include about the hosted zone.
- **private_zone** (*bool*) – Set True if creating a private hosted zone.
- **vpc_id** (*str*) – When creating a private hosted zone, the VPC Id to associate to is required.
- **vpc_region** (*str*) – When creating a private hosted zone, the region of the associated VPC is required.

create_zone (*name*, *private_zone=False*, *vpc_id=None*, *vpc_region=None*)

Create a new Hosted Zone. Returns a Zone object for the newly created Hosted Zone.

Parameters

- **name** (*str*) – The name of the domain. This should be a fully-specified domain, and should end with a final period as the last label indication. If you omit the final period, Amazon Route 53 assumes the domain is relative to the root. This is the name you have registered with your DNS registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request.
- **private_zone** (*bool*) – Set True if creating a private hosted zone.
- **vpc_id** (*str*) – When creating a private hosted zone, the VPC Id to associate to is required.
- **vpc_region** (*str*) – When creating a private hosted zone, the region of the associated VPC is required.

delete_health_check (*health_check_id*)

Delete a health check

Parameters **health_check_id** (*str*) – ID of the health check to delete

delete_hosted_zone (*hosted_zone_id*)

Delete the hosted zone specified by the given id.

Parameters **hosted_zone_id** (*str*) – The hosted zone's id

get_all_hosted_zones (*start_marker=None, zone_list=None*)

Returns a Python data structure with information about all Hosted Zones defined for the AWS account.

Parameters

- **start_marker** (*int*) – start marker to pass when fetching additional results after a truncated list
- **zone_list** (*list*) – a HostedZones list to prepend to results

get_all_rrsets (*hosted_zone_id, type=None, name=None, identifier=None, maxitems=None*)

Retrieve the Resource Record Sets defined for this Hosted Zone. Returns the raw XML data returned by the Route53 call.

Parameters

- **hosted_zone_id** (*str*) – The unique identifier for the Hosted Zone
- **type** (*str*) – The type of resource record set to begin the record listing from.
Valid choices are:

- A
- AAAA
- CNAME
- MX
- NS
- PTR
- SOA
- SPF
- SRV
- TXT

Valid values for weighted resource record sets:

- A
- AAAA
- CNAME
- TXT

Valid values for Zone Apex Aliases:

- A
- AAAA

- **name** (*str*) – The first name in the lexicographic ordering of domain names to be retrieved

- **identifier** (*str*) – In a hosted zone that includes weighted resource record sets (multiple resource record sets with the same DNS name and type that are differentiated only by SetIdentifier), if results were truncated for a given DNS name and type, the value of SetIdentifier for the next resource record set that has the current DNS name and type
- **maxitems** (*int*) – The maximum number of records

get_change (*change_id*)

Get information about a proposed set of changes, as submitted by the `change_rrsets` method. Returns a Python data structure with status information about the changes.

Parameters **change_id** (*str*) – The unique identifier for the set of changes. This ID is returned in the response to the `change_rrsets` method.

get_checker_ip_ranges ()

Return a list of Route53 healthcheck IP ranges

get_hosted_zone (*hosted_zone_id*)

Get detailed information about a particular Hosted Zone.

Parameters **hosted_zone_id** (*str*) – The unique identifier for the Hosted Zone

get_hosted_zone_by_name (*hosted_zone_name*)

Get detailed information about a particular Hosted Zone.

Parameters **hosted_zone_name** (*str*) – The fully qualified domain name for the Hosted Zone

get_list_health_checks (*maxitems=None, marker=None*)

Return a list of health checks

Parameters

- **maxitems** (*int*) – Maximum number of items to return
- **marker** (*str*) – marker to get next set of items to list

get_zone (*name*)

Returns a Zone object for the specified Hosted Zone.

Parameters **name** – The name of the domain. This should be a fully-specified domain, and should end with a final period as the last label indication.

get_zones ()

Returns a list of Zone objects, one for each of the Hosted Zones defined for the AWS account.

Return type *list*

Returns A list of Zone objects.

make_request (*action, path, headers=None, data="", params=None*)

Makes a request to the server, with stock multiple-retry logic.

boto.route53.exception

exception `boto.route53.exception.DNSServerError` (*status, reason, body=None, *args*)

boto.route53.healthcheck

From http://docs.aws.amazon.com/Route53/latest/APIReference/API_CreateHealthCheck.html

POST /2013-04-01/healthcheck HTTP/1.1

```
<?xml version="1.0" encoding="UTF-8"?> <CreateHealthCheckRequest xmlns="https://route53.amazonaws.com/doc/2013-04-01/">
```

```

<CallerReference>unique description</CallerReference> <HealthCheckConfig>
  <IPAddress>IP address of the endpoint to check</IPAddress> <Port>port on the endpoint
  to check</Port> <Type>HTTP | HTTPS | HTTP_STR_MATCH | HTTPS_STR_MATCH |
  TCP</Type> <ResourcePath>path of the file that
    you want Amazon Route 53 to request</ResourcePath>
  <FullyQualifiedDomainName>domain name of the endpoint to
  check</FullyQualifiedDomainName>
  <SearchString>if Type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string
  to search for in the response body from the specified resource</SearchString>
  <RequestInterval>10 | 30</RequestInterval> <FailureThreshold>integer between 1 and
  10</FailureThreshold>
</HealthCheckConfig>

```

```
</CreateHealthCheckRequest>
```

```

class boto.route53.healthcheck.HealthCheck(ip_addr, port, hc_type, resource_path,
                                             fqdn=None, string_match=None, re-
                                             quest_interval=30, failure_threshold=3)

```

An individual health check

HealthCheck object

Parameters

- **ip_addr** (*str*) – Optional IP Address
- **port** (*int*) – Port to check
- **hc_type** (*str*) – One of HTTP | HTTPS | HTTP_STR_MATCH | HTTPS_STR_MATCH | TCP
- **resource_path** (*str*) – Path to check
- **fqdn** (*str*) – domain name of the endpoint to check
- **string_match** (*str*) – if hc_type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to search for in the response body from the specified resource
- **request_interval** (*int*) – The number of seconds between the time that Amazon Route 53 gets a response from your endpoint and the time that it sends the next health-check request.
- **failure_threshold** (*int*) – The number of consecutive health checks that an endpoint must pass or fail for Amazon Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

```
POSTXMLBody = '\n <HealthCheckConfig>\n %(ip_addr_part)s\n <Port>%(port)s</Port>\n <Ty
```

```
XMLFQDNPart = '<FullyQualifiedDomainName>%(fqdn)s</FullyQualifiedDomainName>'
```

```
XMLIpAddrPart = '<IPAddress>%(ip_addr)s</IPAddress>'
```

```
XMLRequestIntervalPart = '<RequestInterval>%(request_interval)d</RequestInterval>'
```

```
XMLStringMatchPart = '<SearchString>%(string_match)s</SearchString>'
```

```
to_xml()
```

```
valid_request_intervals = (10, 30)
```

boto.route53.hostedzone

```
class boto.route53.hostedzone.HostedZone (id=None, name=None, owner=None, version=None, caller_reference=None)
```

```
endElement (name, value, connection)
```

```
startElement (name, attrs, connection)
```

boto.route53.record

```
class boto.route53.record.Record (name=None, type=None, ttl=600, resource_records=None, alias_hosted_zone_id=None, alias_dns_name=None, identifier=None, weight=None, region=None, alias_evaluate_target_health=None, health_check=None, failover=None)
```

An individual ResourceRecordSet

```
AliasBody = '<AliasTarget>\n <HostedZoneId>% (hosted_zone_id) s</HostedZoneId>\n <DNSName>% (dns_name) s</DNSName>'
```

```
EvaluateTargetHealth = '<EvaluateTargetHealth>% s</EvaluateTargetHealth>'
```

```
FailoverBody = '\n <SetIdentifier>% (identifier) s</SetIdentifier>\n <Failover>% (failover) s</Failover>'
```

```
HealthCheckBody = '<HealthCheckId>% s</HealthCheckId>'
```

```
RRRBody = '\n <SetIdentifier>% (identifier) s</SetIdentifier>\n <Region>% (region) s</Region>'
```

```
ResourceRecordBody = '<ResourceRecord>\n <Value>% s</Value>\n </ResourceRecord>'
```

```
ResourceRecordsBody = '\n <TTL>% (ttl) s</TTL>\n <ResourceRecords>\n % (records) s</ResourceRecords>'
```

```
WRRBody = '\n <SetIdentifier>% (identifier) s</SetIdentifier>\n <Weight>% (weight) s</Weight>'
```

```
XMLBody = '<ResourceRecordSet>\n <Name>% (name) s</Name>\n <Type>% (type) s</Type>\n % (weight) s</ResourceRecordSet>'
```

```
add_value (value)
```

Add a resource record value

```
endElement (name, value, connection)
```

```
set_alias (alias_hosted_zone_id, alias_dns_name, alias_evaluate_target_health=False)
```

Make this an alias resource record set

```
startElement (name, attrs, connection)
```

```
to_print ()
```

```
to_xml ()
```

Spit this resource record set out as XML

```
class boto.route53.record.ResourceRecordSets (connection=None, hosted_zone_id=None, comment=None)
```

A list of resource records.

Variables

- **hosted_zone_id** – The ID of the hosted zone.
- **comment** – A comment that will be stored with the change.
- **changes** – A list of changes.

```
ChangeResourceRecordSetsBody = '<?xml version="1.0" encoding="UTF-8"?>\n <ChangeResourceRecordSets>'
```

```
ChangeXML = '<Change>\n <Action>% (action) s</Action>\n % (record) s</Change>'
```

add_change (*action, name, type, ttl=600, alias_hosted_zone_id=None, alias_dns_name=None, identifier=None, weight=None, region=None, alias_evaluate_target_health=None, health_check=None, failover=None*)

Add a change request to the set.

Parameters

- **action** (*str*) – The action to perform ('CREATE'|'DELETE'|'UPSERT')
- **name** (*str*) – The name of the domain you want to perform the action on.
- **type** (*str*) – The DNS record type. Valid values are:
 - A
 - AAAA
 - CNAME
 - MX
 - NS
 - PTR
 - SOA
 - SPF
 - SRV
 - TXT
- **ttl** (*int*) – The resource record cache time to live (TTL), in seconds.
- **alias_dns_name** (*str*) – *Alias resource record sets only* The value of the hosted zone ID, CanonicalHostedZoneNameId, for the LoadBalancer.
- **alias_hosted_zone_id** (*str*) – *Alias resource record sets only* Information about the domain to which you are redirecting traffic.
- **identifier** (*str*) – *Weighted and latency-based resource record sets only* An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type.
- **weight** (*int*) – *Weighted resource record sets only* Among resource record sets that have the same combination of DNS name and type, a value that determines what portion of traffic for the current resource record set is routed to the associated location
- **region** (*str*) – *Latency-based resource record sets only* Among resource record sets that have the same combination of DNS name and type, a value that determines which region this should be associated with for the latency-based routing
- **alias_evaluate_target_health** (*bool*) – *Required for alias resource record sets* Indicates whether this Resource Record Set should respect the health status of any health checks associated with the ALIAS target record which it is linked to.
- **health_check** (*str*) – Health check to associate with this record
- **failover** (*str*) – *Failover resource record sets only* Whether this is the primary or secondary resource record set.

add_change_record (*action, change*)

Add an existing record to a change set with the specified action

commit ()
Commit this change

endElement (*name, value, connection*)
Overwritten to also add the NextRecordName, NextRecordType and NextRecordIdentifier to the base object

to_xml ()
Convert this ResourceRecordSet into XML to be saved via the ChangeResourceRecordSetsRequest

boto.route53.status

class boto.route53.status.**Status** (*route53connection, change_dict*)

update ()
Update the status of this request.

boto.route53.zone

class boto.route53.zone.**Zone** (*route53connection, zone_dict*)

A Route53 Zone.

Variables

- **route53connection** – A *boto.route53.connection.Route53Connection* connection
- **id** – The ID of the hosted zone

add_a (*name, value, ttl=None, identifier=None, comment=""*)
Add a new A record to this Zone. See `_new_record` for parameter documentation. Returns a Status object.

add_cname (*name, value, ttl=None, identifier=None, comment=""*)
Add a new CNAME record to this Zone. See `_new_record` for parameter documentation. Returns a Status object.

add_mx (*name, records, ttl=None, identifier=None, comment=""*)
Add a new MX record to this Zone. See `_new_record` for parameter documentation. Returns a Status object.

add_record (*resource_type, name, value, ttl=60, identifier=None, comment=""*)
Add a new record to this Zone. See `_new_record` for parameter documentation. Returns a Status object.

delete ()
Request that this zone be deleted by Amazon.

delete_a (*name, identifier=None, all=False*)
Delete an A record matching name and identifier from this Zone. Returns a Status object.

If there is more than one match delete all matching records if `all` is `True`, otherwise throws `TooManyRecordsException`.

delete_cname (*name, identifier=None, all=False*)
Delete a CNAME record matching name and identifier from this Zone. Returns a Status object.

If there is more than one match delete all matching records if `all` is `True`, otherwise throws `TooManyRecordsException`.

delete_mx (*name, identifier=None, all=False*)
Delete an MX record matching name and identifier from this Zone. Returns a Status object.

If there is more than one match delete all matching records if `all` is `True`, otherwise throws `TooManyRecordsException`.

delete_record (*record*, *comment=""*)

Delete one or more records from this Zone. Returns a Status object.

Parameters

- **record** – A ResourceRecord (e.g. returned by `find_records`) or list, tuple, or set of ResourceRecords.
- **comment** (*str*) – A comment that will be stored with the change.

find_records (*name*, *type*, *desired=1*, *all=False*, *identifier=None*)

Search this Zone for records that match given parameters. Returns `None` if no results, a `ResourceRecord` if one result, or a `ResourceRecordSets` if more than one result.

Parameters

- **name** (*str*) – The name of the records should match this parameter
- **type** (*str*) – The type of the records should match this parameter
- **desired** (*int*) – The number of desired results. If the number of matching records in the Zone exceeds the value of this parameter, throw `TooManyRecordsException`
- **all** (*Boolean*) – If true return all records that match name, type, and identifier parameters
- **identifier** (*Tuple*) – A tuple specifying WRR or LBR attributes. Valid forms are:
 - (str, int): WRR record [e.g. ('foo',10)]
 - (str, str): LBR record [e.g. ('foo','us-east-1')]

get_a (*name*, *all=False*)

Search this Zone for A records that match name.

Returns a ResourceRecord.

If there is more than one match return all as a `ResourceRecordSets` if `all` is `True`, otherwise throws `TooManyRecordsException`.

get_cname (*name*, *all=False*)

Search this Zone for CNAME records that match name.

Returns a ResourceRecord.

If there is more than one match return all as a `ResourceRecordSets` if `all` is `True`, otherwise throws `TooManyRecordsException`.

get_mx (*name*, *all=False*)

Search this Zone for MX records that match name.

Returns a ResourceRecord.

If there is more than one match return all as a `ResourceRecordSets` if `all` is `True`, otherwise throws `TooManyRecordsException`.

get_nameservers ()

Get the list of nameservers for this zone.

get_records ()

Return a `ResourceRecordsSets` for all of the records in this zone.

update_a (*name, value, ttl=None, identifier=None, comment=""*)
 Update the given A record in this Zone to a new value, ttl, and identifier. Returns a Status object.
 Will throw TooManyRecordsException is name, value does not match a single record.

update_cname (*name, value, ttl=None, identifier=None, comment=""*)
 Update the given CNAME record in this Zone to a new value, ttl, and identifier. Returns a Status object.
 Will throw TooManyRecordsException is name, value does not match a single record.

update_mx (*name, value, ttl=None, identifier=None, comment=""*)
 Update the given MX record in this Zone to a new value, ttl, and identifier. Returns a Status object.
 Will throw TooManyRecordsException is name, value does not match a single record.

update_record (*old_record, new_value, new_ttl=None, new_identifier=None, comment=""*)
 Update an existing record in this Zone. Returns a Status object.
Parameters **old_record** (*ResourceRecord*) – A ResourceRecord (e.g. returned by `find_records`)
 See `_new_record` for additional parameter documentation.

4.138.9 S3

boto.s3

class `boto.s3.S3RegionInfo` (*connection=None, name=None, endpoint=None, connection_cls=None*)

connect (***kw_params*)

Connect to this Region's endpoint. Returns an connection object pointing to the endpoint associated with this region. You may pass any of the arguments accepted by the connection class's constructor as keyword arguments and they will be passed along to the connection object.

Return type Connection object

Returns The connection to this regions endpoint

`boto.s3.connect_to_region` (*region_name, **kw_params*)

`boto.s3.regions` ()

Get all available regions for the Amazon S3 service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

boto.s3.acl

class `boto.s3.acl.ACL` (*policy=None*)

add_email_grant (*permission, email_address*)

add_grant (*grant*)

add_user_grant (*permission, user_id, display_name=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

```
class boto.s3.acl.Grant (permission=None, type=None, id=None, display_name=None, uri=None, email_address=None)
```

```
    Namespace = 'xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"'
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
    to_xml ()
```

```
class boto.s3.acl.Policy (parent=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
    to_xml ()
```

boto.s3.bucket

```
class boto.s3.bucket.Bucket (connection=None, name=None, key_class=<class 'boto.s3.key.Key'>)
```

```
    BucketPaymentBody = '<?xml version="1.0" encoding="UTF-8"?>\n <RequestPaymentConfigura
```

```
    LoggingGroup = 'http://acs.amazonaws.com/groups/s3/LogDelivery'
```

```
    MFADeleteRE = '<MfaDelete> ([A-Za-z]+) </MfaDelete>'
```

```
    VersionRE = '<Status> ([A-Za-z]+) </Status>'
```

```
    VersioningBody = '<?xml version="1.0" encoding="UTF-8"?>\n <VersioningConfiguration xm
```

```
    add_email_grant (permission, email_address, recursive=False, headers=None)
```

Convenience method that provides a quick way to add an email grant to a bucket. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to S3.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ, WRITE, READ_ACP, WRITE_ACP, FULL_CONTROL).
- **email_address** (*string*) – The email address associated with the AWS account you are granting the permission to.
- **recursive** (*boolean*) – A boolean value to controls whether the command will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!

```
    add_user_grant (permission, user_id, recursive=False, headers=None, display_name=None)
```

Convenience method that provides a quick way to add a canonical user grant to a bucket. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to S3.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ, WRITE, READ_ACP, WRITE_ACP, FULL_CONTROL).

- **user_id** (*string*) – The canonical user id associated with the AWS account your are granting the permission to.
- **recursive** (*boolean*) – A boolean value to controls whether the command will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!
- **display_name** (*string*) – An option string containing the user’s Display Name. Only required on Walrus.

cancel_multipart_upload (*key_name, upload_id, headers=None*)

To verify that all parts have been removed, so you don’t get charged for the part storage, you should call the List Parts operation and ensure the parts list is empty.

complete_multipart_upload (*key_name, upload_id, xml_body, headers=None*)

Complete a multipart upload operation.

configure_lifecycle (*lifecycle_config, headers=None*)

Configure lifecycle for this bucket.

Parameters lifecycle_config (*boto.s3.lifecycle.Lifecycle*) – The lifecycle configuration you want to configure for this bucket.

configure_versioning (*versioning, mfa_delete=False, mfa_token=None, headers=None*)

Configure versioning for this bucket.

..note:: This feature is currently in beta.

Parameters

- **versioning** (*bool*) – A boolean indicating whether version is enabled (True) or disabled (False).
- **mfa_delete** (*bool*) – A boolean indicating whether the Multi-Factor Authentication Delete feature is enabled (True) or disabled (False). If mfa_delete is enabled then all Delete operations will require the token from your MFA device to be passed in the request.
- **mfa_token** (*tuple or list of strings*) – A tuple or list consisting of the serial number from the MFA device and the current value of the six-digit token associated with the device. This value is required when you are changing the status of the MfaDelete property of the bucket.

configure_website (*suffix=None, error_key=None, redirect_all_requests_to=None, routing_rules=None, headers=None*)

Configure this bucket to act as a website

Parameters

- **suffix** (*str*) – Suffix that is appended to a request that is for a “directory” on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html). The suffix must not be empty and must not include a slash character.
- **error_key** (*str*) – The object key name to use when a 4XX class error occurs. This is optional.
- **redirect_all_requests_to** (*boto.s3.website.RedirectLocation*) – Describes the redirect behavior for every request to this bucket’s website endpoint. If this value is non None, no other values are

considered when configuring the website configuration for the bucket. This is an instance of `RedirectLocation`.

- **routing_rules** (`boto.s3.website.RoutingRules`) – Object which specifies conditions and redirects that apply when the conditions are met.

copy_key (*new_key_name*, *src_bucket_name*, *src_key_name*, *metadata=None*, *src_version_id=None*, *storage_class='STANDARD'*, *preserve_acl=False*, *encrypt_key=False*, *headers=None*, *query_args=None*)

Create a new key in the bucket by copying another existing key.

Parameters

- **new_key_name** (*string*) – The name of the new key
- **src_bucket_name** (*string*) – The name of the source bucket
- **src_key_name** (*string*) – The name of the source key
- **src_version_id** (*string*) – The version id for the key. This param is optional. If not specified, the newest version of the key will be copied.
- **metadata** (*dict*) – Metadata to be associated with new key. If metadata is supplied, it will replace the metadata of the source key being copied. If no metadata is supplied, the source key's metadata will be copied to the new key.
- **storage_class** (*string*) – The storage class of the new key. By default, the new key will use the standard storage class. Possible values are: STANDARD | REDUCED_REDUNDANCY
- **preserve_acl** (*bool*) – If True, the ACL from the source key will be copied to the destination key. If False, the destination key will have the default ACL. Note that preserving the ACL in the new key object will require two additional API calls to S3, one to retrieve the current ACL and one to set that ACL on the new object. If you don't care about the ACL, a value of False will be significantly more efficient.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **headers** (*dict*) – A dictionary of header name/value pairs.
- **query_args** (*string*) – A string of additional querystring arguments to append to the request

Return type `boto.s3.key.Key` or subclass

Returns An instance of the newly created key object

delete (*headers=None*)

delete_cors (*headers=None*)

Removes all CORS configuration from the bucket.

delete_key (*key_name*, *headers=None*, *version_id=None*, *mfa_token=None*)

Deletes a key from the bucket. If a *version_id* is provided, only that version of the key will be deleted.

Parameters

- **key_name** (*string*) – The key name to delete
- **version_id** (*string*) – The version ID (optional)
- **mfa_token** (*tuple or list of strings*) – A tuple or list consisting of the serial number from the MFA device and the current value of the six-digit

token associated with the device. This value is required anytime you are deleting versioned objects from a bucket that has the MFADelete option on the bucket.

Return type `boto.s3.key.Key` or subclass

Returns A key object holding information on what was deleted. The Caller can see if a `delete_marker` was created or removed and what `version_id` the delete created or removed.

delete_keys (*keys, quiet=False, mfa_token=None, headers=None*)

Deletes a set of keys using S3's Multi-object delete API. If a VersionID is specified for that key then that version is removed. Returns a MultiDeleteResult Object, which contains Deleted and Error elements for each key you ask to delete.

Parameters

- **keys** (*list*) – A list of either `key_names` or (`key_name`, `versionid`) pairs or a list of `Key` instances.
- **quiet** (*boolean*) – In quiet mode the response includes only keys where the delete operation encountered an error. For a successful deletion, the operation does not return any information about the delete in the response body.
- **mfa_token** (*tuple or list of strings*) – A tuple or list consisting of the serial number from the MFA device and the current value of the six-digit token associated with the device. This value is required anytime you are deleting versioned objects from a bucket that has the MFADelete option on the bucket.

Returns An instance of `MultiDeleteResult`

delete_lifecycle_configuration (*headers=None*)

Removes all lifecycle configuration from the bucket.

delete_policy (*headers=None*)

delete_tags (*headers=None*)

delete_website_configuration (*headers=None*)

Removes all website configuration from the bucket.

disable_logging (*headers=None*)

Disable logging on a bucket.

Return type `bool`

Returns True if ok or raises an exception.

enable_logging (*target_bucket, target_prefix="", grants=None, headers=None*)

Enable logging on a bucket.

Parameters

- **target_bucket** (*bucket or string*) – The bucket to log to.
- **target_prefix** (*string*) – The prefix which should be prepended to the generated log files written to the `target_bucket`.
- **grants** (*list of Grant objects*) – A list of extra permissions which will be granted on the log files which are created.

Return type `bool`

Returns True if ok or raises an exception.

endElement (*name, value, connection*)

generate_url (*expires_in, method='GET', headers=None, force_http=False, response_headers=None, expires_in_absolute=False*)

get_acl (*key_name=""*, *headers=None*, *version_id=None*)

get_all_keys (*headers=None*, ***params*)

A lower-level method for listing contents of a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the list method.

Parameters

- **max_keys** (*int*) – The maximum number of keys to retrieve
- **prefix** (*string*) – The prefix of the keys you want to retrieve
- **marker** (*string*) – The “marker” of where you are in the result set
- **delimiter** (*string*) – If this optional, Unicode string parameter is included with your request, then keys that contain the same string between the prefix and the first occurrence of the delimiter will be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

Return type *ResultSet*

Returns The result from S3 listing the keys requested

get_all_multipart_uploads (*headers=None*, ***params*)

A lower-level, version-aware method for listing active MultiPart uploads for a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the list method.

Parameters

- **max_uploads** (*int*) – The maximum number of uploads to retrieve. Default value is 1000.
- **key_marker** (*string*) – Together with `upload_id_marker`, this parameter specifies the multipart upload after which listing should begin. If `upload_id_marker` is not specified, only the keys lexicographically greater than the specified `key_marker` will be included in the list.

If `upload_id_marker` is specified, any multipart uploads for a key equal to the `key_marker` might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified `upload_id_marker`.
- **upload_id_marker** (*string*) – Together with `key-marker`, specifies the multipart upload after which listing should begin. If `key_marker` is not specified, the `upload_id_marker` parameter is ignored. Otherwise, any multipart uploads for a key equal to the `key_marker` might be included in the list only if they have an upload ID lexicographically greater than the specified `upload_id_marker`.
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

- **delimiter** (*string*) – Character you use to group keys. All keys that contain the same string between the prefix, if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, `CommonPrefixes`. If you don't specify the prefix parameter, then the substring starts at the beginning of the key. The keys that are grouped under `CommonPrefixes` result element are not returned elsewhere in the response.
- **prefix** (*string*) – Lists in-progress uploads only for those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different grouping of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)

Return type *ResultSet*

Returns The result from S3 listing the uploads requested

get_all_versions (*headers=None, **params*)

A lower-level, version-aware method for listing contents of a bucket. This closely models the actual S3 API and requires you to manually handle the paging of results. For a higher-level method that handles the details of paging for you, you can use the `list` method.

Parameters

- **max_keys** (*int*) – The maximum number of keys to retrieve
- **prefix** (*string*) – The prefix of the keys you want to retrieve
- **key_marker** (*string*) – The “marker” of where you are in the result set with respect to keys.
- **version_id_marker** (*string*) – The “marker” of where you are in the result set with respect to version-id's.
- **delimiter** (*string*) – If this optional, Unicode string parameter is included with your request, then keys that contain the same string between the prefix and the first occurrence of the delimiter will be rolled up into a single result element in the `CommonPrefixes` collection. These rolled-up keys are not returned elsewhere in the response.
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: `url`

Return type *ResultSet*

Returns The result from S3 listing the keys requested

get_cors (*headers=None*)

Returns the current CORS configuration on the bucket.

Return type *boto.s3.cors.CORSConfiguration*

Returns A CORSConfiguration object that describes all current CORS rules in effect for the bucket.

get_cors_xml (*headers=None*)

Returns the current CORS configuration on the bucket as an XML document.

get_key (*key_name, headers=None, version_id=None, response_headers=None, validate=True*)

Check to see if a particular key exists within the bucket. This method uses a HEAD request to check for the existence of the key. Returns: An instance of a Key object or None

Parameters

- **key_name** (*string*) – The name of the key to retrieve
- **headers** (*dict*) – The headers to send when retrieving the key
- **version_id** (*string*) –
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **validate** (*bool*) – Verifies whether the key exists. If `False`, this will not hit the service, constructing an in-memory object. Default is `True`.

Return type `boto.s3.key.Key`

Returns A Key object from this bucket.

get_lifecycle_config (*headers=None*)

Returns the current lifecycle configuration on the bucket.

Return type `boto.s3.lifecycle.Lifecycle`

Returns A LifecycleConfig object that describes all current lifecycle rules in effect for the bucket.

get_location (*headers=None*)

Returns the LocationConstraint for the bucket.

Return type `str`

Returns The LocationConstraint for the bucket or the empty string if no constraint was specified when bucket was created.

get_logging_status (*headers=None*)

Get the logging status for this bucket.

Return type `boto.s3.bucketlogging.BucketLogging`

Returns A BucketLogging object for this bucket.

get_policy (*headers=None*)

Returns the JSON policy associated with the bucket. The policy is returned as an uninterpreted JSON string.

get_request_payment (*headers=None*)

get_subresource (*subresource, key_name="", headers=None, version_id=None*)

Get a subresource for a bucket or key.

Parameters

- **subresource** (*string*) – The subresource to get.
- **key_name** (*string*) – The key to operate on, or `None` to operate on the bucket.
- **headers** (*dict*) – Additional HTTP headers to include in the request.

- **src_version_id** (*string*) – Optional. The version id of the key to operate on. If not specified, operate on the newest version.

Return type string

Returns The value of the subresource.

get_tags (*headers=None*)

get_versioning_status (*headers=None*)

Returns the current status of versioning on the bucket.

Return type dict

Returns A dictionary containing a key named ‘Versioning’ that can have a value of either Enabled, Disabled, or Suspended. Also, if MFADelete has ever been enabled on the bucket, the dictionary will contain a key named ‘MFADelete’ which will have a value of either Enabled or Suspended.

get_website_configuration (*headers=None*)

Returns the current status of website configuration on the bucket.

Return type dict

Returns A dictionary containing a Python representation of the XML response from S3. The overall structure is:

- WebsiteConfiguration
 - IndexDocument
 - * Suffix : suffix that is appended to request that is for a “directory” on the website endpoint
 - * ErrorDocument
 - Key : name of object to serve when an error occurs

get_website_configuration_obj (*headers=None*)

Get the website configuration as a `boto.s3.website.WebsiteConfiguration` object.

get_website_configuration_with_xml (*headers=None*)

Returns the current status of website configuration on the bucket as unparsed XML.

Return type 2-Tuple

Returns

2-tuple containing:

1. A dictionary containing a Python representation of the XML response. The overall structure is:

- WebsiteConfiguration
 - IndexDocument
 - * Suffix : suffix that is appended to request that is for a “directory” on the website endpoint
 - * ErrorDocument
 - Key : name of object to serve when an error occurs

2. unparsed XML describing the bucket’s website configuration

get_website_configuration_xml (*headers=None*)

Get raw website configuration xml

get_website_endpoint ()

Returns the fully qualified hostname to use if you want to access this bucket as a website. This doesn't validate whether the bucket has been correctly configured as a website or not.

get_xml_acl (*key_name=""*, *headers=None*, *version_id=None*)

get_xml_tags (*headers=None*)

initiate_multipart_upload (*key_name*, *headers=None*, *reduced_redundancy=False*, *metadata=None*, *encrypt_key=False*, *policy=None*)

Start a multipart upload operation.

Note: Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Parameters

- **key_name** (*string*) – The name of the key that will ultimately result from this multipart upload operation. This will be exactly as the key appears in the bucket after the upload process has been completed.
- **headers** (*dict*) – Additional HTTP headers to send and store with the resulting key in S3.
- **reduced_redundancy** (*boolean*) – In multipart uploads, the storage class is specified when initiating the upload, not when uploading individual parts. So if you want the resulting key to use the reduced redundancy storage class set this flag when you initiate the upload.
- **metadata** (*dict*) – Any metadata that you would like to set on the key that results from the multipart upload.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **policy** (*boto.s3.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key (once completed) in S3.

list (*prefix=""*, *delimiter=""*, *marker=""*, *headers=None*, *encoding_type=None*)

List key objects within a bucket. This returns an instance of an `BucketListResultSet` that automatically handles all of the result paging, etc. from S3. You just need to keep iterating until there are no more results.

Called with no arguments, this will return an iterator object across all keys within the bucket.

The Key objects returned by the iterator are obtained by parsing the results of a GET on the bucket, also known as the List Objects request. The XML returned by this request contains only a subset of the information about each key. Certain metadata fields such as Content-Type and user metadata are not available in the XML. Therefore, if you want these additional metadata fields you will have to do a HEAD request on the Key in the bucket.

Parameters

- **prefix** (*string*) – allows you to limit the listing to a particular prefix. For example, if you call the method with `prefix='/foo/'` then the iterator will only cycle through the keys that begin with the string `'foo/'`.

- **delimiter** (*string*) – can be used in conjunction with the prefix to allow you to organize and browse your keys hierarchically. See <http://goo.gl/Xx63h> for more details.
- **marker** (*string*) – The “marker” of where you are in the result set
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: url

Return type *boto.s3.bucketlistresultset.BucketListResultSet*

Returns an instance of a BucketListResultSet that handles paging, etc

list_grants (*headers=None*)

list_multipart_uploads (*key_marker="", upload_id_marker="", headers=None, encoding_type=None*)

List multipart upload objects within a bucket. This returns an instance of an MultiPartUploadListResultSet that automatically handles all of the result paging, etc. from S3. You just need to keep iterating until there are no more results.

Parameters

- **key_marker** (*string*) – The “marker” of where you are in the result set
- **upload_id_marker** (*string*) – The upload identifier
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: url

Return type *boto.s3.bucketlistresultset.MultiPartUploadListResultSet*

Returns an instance of a BucketListResultSet that handles paging, etc

list_versions (*prefix="", delimiter="", key_marker="", version_id_marker="", headers=None, encoding_type=None*)

List version objects within a bucket. This returns an instance of an VersionedBucketListResultSet that automatically handles all of the result paging, etc. from S3. You just need to keep iterating until there are no more results. Called with no arguments, this will return an iterator object across all keys within the bucket.

Parameters

- **prefix** (*string*) – allows you to limit the listing to a particular prefix. For example, if you call the method with prefix='/foo/' then the iterator will only cycle through the keys that begin with the string '/foo/'.
- **delimiter** (*string*) – can be used in conjunction with the prefix to allow you to organize and browse your keys hierarchically. See:

<http://aws.amazon.com/releasenotes/Amazon-S3/213>

for more details.

- **key_marker** (*string*) – The “marker” of where you are in the result set
- **encoding_type** (*string*) – Requests Amazon S3 to encode the response and specifies the encoding method to use.

An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid options: url

Return type *boto.s3.bucketlistresultset.BucketListResultSet*

Returns an instance of a BucketListResultSet that handles paging, etc

lookup (*key_name*, *headers=None*)

Deprecated: Please use get_key method.

Parameters **key_name** (*string*) – The name of the key to retrieve

Return type *boto.s3.key.Key*

Returns A Key object from this bucket.

make_public (*recursive=False*, *headers=None*)

new_key (*key_name=None*)

Creates a new key

Parameters **key_name** (*string*) – The name of the key to create

Return type *boto.s3.key.Key* or subclass

Returns An instance of the newly created key object

set_acl (*acl_or_str*, *key_name=""*, *headers=None*, *version_id=None*)

set_as_logging_target (*headers=None*)

Setup the current bucket as a logging target by granting the necessary permissions to the LogDelivery group to write log files to this bucket.

set_canned_acl (*acl_str*, *key_name=""*, *headers=None*, *version_id=None*)

set_cors (*cors_config*, *headers=None*)

Set the CORS for this bucket given a boto CORSConfiguration object.

Parameters **cors_config** (*boto.s3.cors.CORSConfiguration*) – The CORS configuration you want to configure for this bucket.

set_cors_xml (*cors_xml*, *headers=None*)

Set the CORS (Cross-Origin Resource Sharing) for a bucket.

Parameters **cors_xml** (*str*) – The XML document describing your desired CORS configuration. See the S3 documentation for details of the exact syntax required.

set_key_class (*key_class*)

Set the Key class associated with this bucket. By default, this would be the boto.s3.key.Key class but if you want to subclass that for some reason this allows you to associate your new class with a bucket so that when you call bucket.new_key() or when you get a listing of keys in the bucket you will get an instances of your key class rather than the default.

Parameters **key_class** (*class*) – A subclass of Key that can be more specific

set_policy (*policy*, *headers=None*)

Add or replace the JSON policy associated with the bucket.

Parameters **policy** (*str*) – The JSON policy as a string.

set_request_payment (*payer='BucketOwner', headers=None*)

set_subresource (*subresource, value, key_name="", headers=None, version_id=None*)
Set a subresource for a bucket or key.

Parameters

- **subresource** (*string*) – The subresource to set.
- **value** (*string*) – The value of the subresource.
- **key_name** (*string*) – The key to operate on, or None to operate on the bucket.
- **headers** (*dict*) – Additional HTTP headers to include in the request.
- **src_version_id** (*string*) – Optional. The version id of the key to operate on. If not specified, operate on the newest version.

set_tags (*tags, headers=None*)

set_website_configuration (*config, headers=None*)

Parameters config (*boto.s3.website.WebsiteConfiguration*) – Configuration data

set_website_configuration_xml (*xml, headers=None*)

Upload xml website configuration

set_xml_acl (*acl_str, key_name="", headers=None, version_id=None, query_args='acl'*)

set_xml_logging (*logging_str, headers=None*)

Set logging on a bucket directly to the given xml string.

Parameters logging_str (*unicode string*) – The XML for the bucket logging status which will be set. The string will be converted to utf-8 before it is sent. Usually, you will obtain this XML from the BucketLogging object.

Return type `bool`

Returns True if ok or raises an exception.

set_xml_tags (*tag_str, headers=None, query_args='tagging'*)

startElement (*name, attrs, connection*)

validate_get_all_versions_params (*params*)

Validate that the parameters passed to `get_all_versions` are valid. Overridden by subclasses that allow a different set of parameters.

Parameters params (*dict*) – Parameters to validate.

validate_kwarg_names (*kwargs, names*)

Checks that all named arguments are in the specified list of names.

Parameters

- **kwargs** (*dict*) – Dictionary of kwargs to validate.
- **names** (*list*) – List of possible named arguments.

class `boto.s3.bucket.S3WebsiteEndpointTranslate`

`trans_region = {'ap-northeast-1': 's3-website-ap-northeast-1', 'ap-southeast-1': 's3-`

`classmethod translate_region (reg)`

boto.s3.bucketlistresultset

```
class boto.s3.bucketlistresultset.BucketListResultSet (bucket=None, prefix="",  
delimiter="", marker="",  
headers=None, encod-  
ing_type=None)
```

A resultset for listing keys within a bucket. Uses the `bucket_lister` generator function and implements the iterator interface. This transparently handles the results paging from S3 so even if you have many thousands of keys within the bucket you can iterate over all keys in a reasonably efficient manner.

```
class boto.s3.bucketlistresultset.MultiPartUploadListResultSet (bucket=None,  
key_marker="",  
up-  
load_id_marker="",  
headers=None,  
encod-  
ing_type=None)
```

A resultset for listing multipart uploads within a bucket. Uses the `multipart_upload_lister` generator function and implements the iterator interface. This transparently handles the results paging from S3 so even if you have many thousands of uploads within the bucket you can iterate over all keys in a reasonably efficient manner.

```
class boto.s3.bucketlistresultset.VersionedBucketListResultSet (bucket=None,  
prefix="", de-  
limiter="",  
key_marker="",  
ver-  
sion_id_marker="",  
headers=None,  
encod-  
ing_type=None)
```

A resultset for listing versions within a bucket. Uses the `bucket_lister` generator function and implements the iterator interface. This transparently handles the results paging from S3 so even if you have many thousands of keys within the bucket you can iterate over all keys in a reasonably efficient manner.

```
boto.s3.bucketlistresultset.bucket_lister (bucket, prefix="", delimiter="", marker="", head-  
ers=None, encoding_type=None)
```

A generator function for listing keys in a bucket.

```
boto.s3.bucketlistresultset.multipart_upload_lister (bucket, key_marker="",  
upload_id_marker="",  
headers=None, encod-  
ing_type=None)
```

A generator function for listing multipart uploads in a bucket.

```
boto.s3.bucketlistresultset.versioned_bucket_lister (bucket, prefix="", delim-  
iter="", key_marker="",  
version_id_marker="",  
headers=None, encod-  
ing_type=None)
```

A generator function for listing versions in a bucket.

boto.s3.connection

```
exception boto.s3.connection.HostRequiredError (reason, *args)
```

```
class boto.s3.connection.Location
```

```

APNortheast = 'ap-northeast-1'
APSEast = 'ap-southeast-1'
APSEast2 = 'ap-southeast-2'
CNNorth1 = 'cn-north-1'
DEFAULT = ''
EU = 'EU'
EUCentral1 = 'eu-central-1'
SAEast = 'sa-east-1'
USWest = 'us-west-1'
USWest2 = 'us-west-2'

class boto.s3.connection.NoHostProvided
class boto.s3.connection.OrdinaryCallingFormat

    build_path_base (bucket, key="")
    get_bucket_server (server, bucket)

class boto.s3.connection.ProtocolIndependentOrdinaryCallingFormat

    build_url_base (connection, protocol, server, bucket, key="")

class boto.s3.connection.S3Connection (aws_access_key_id=None,
                                         aws_secret_access_key=None, is_secure=True,
                                         port=None, proxy=None, proxy_port=None,
                                         proxy_user=None, proxy_pass=None, host=<class
                                         'boto.s3.connection.NoHostProvided'>, de-
                                         bug=0, https_connection_factory=None, call-
                                         ing_format='boto.s3.connection.SubdomainCallingFormat',
                                         path='/', provider='aws', bucket_class=<class
                                         'boto.s3.bucket.Bucket'>, security_token=None,
                                         suppress_consec_slashes=True, anon=None, vali-
                                         date_certs=None, profile_name=None)

DefaultCallingFormat = 'boto.s3.connection.SubdomainCallingFormat'
DefaultHost = 's3.amazonaws.com'
QueryString = 'Signature=%s&Expires=%d&AWSAccessKeyId=%s'
build_post_form_args (bucket_name, key, expires_in=6000, acl=None, suc-
                      cess_action_redirect=None, max_content_length=None,
                      http_method='http', fields=None, conditions=None, stor-
                      age_class='STANDARD', server_side_encryption=None)

```

Taken from the AWS book Python examples and modified for use with boto This only returns the arguments required for the post form, not the actual form. This does not return the file input field which also needs to be added

Parameters

- **bucket_name** (*string*) – Bucket to submit to
- **key** (*string*) – Key name, optionally add `${filename}` to the end to attach the submitted filename

- **expires_in** (*integer*) – Time (in seconds) before this expires, defaults to 6000
- **acl** (*string*) – A canned ACL. One of: * private * public-read * public-read-write * authenticated-read * bucket-owner-read * bucket-owner-full-control
- **success_action_redirect** (*string*) – URL to redirect to on success
- **max_content_length** (*integer*) – Maximum size for this file
- **http_method** (*string*) – HTTP Method to use, “http” or “https”
- **storage_class** (*string*) – Storage class to use for storing the object. Valid values: STANDARD | REDUCED_REDUNDANCY
- **server_side_encryption** (*string*) – Specifies server-side encryption algorithm to use when Amazon S3 creates an object. Valid values: None | AES256

Return type dict

Returns

A dictionary containing field names/values as well as a url to POST to



build_post_policy (*expiration_time, conditions*)

Taken from the AWS book Python examples and modified for use with boto

create_bucket (*bucket_name, headers=None, location="", policy=None*)

Creates a new located bucket. By default it's in the USA. You can pass Location.EU to create a European bucket (S3) or European Union bucket (GCS).

Parameters

- **bucket_name** (*string*) – The name of the new bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **location** (*str*) – The location of the new bucket. You can use one of the constants in *boto.s3.connection.Location* (e.g. Location.EU, Location.USWest, etc.).
- **policy** (*boto.s3.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key in S3.

delete_bucket (*bucket, headers=None*)

Removes an S3 bucket.

In order to remove the bucket, it must first be empty. If the bucket is not empty, an *S3ResponseError* will be raised.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.

generate_url (*expires_in, method, bucket="", key="", headers=None, query_auth=True, force_http=False, response_headers=None, expires_in_absolute=False, version_id=None*)

generate_url_sigv4 (*expires_in, method, bucket="", key="", headers=None, force_http=False, response_headers=None, version_id=None, iso_date=None*)

get_all_buckets (*headers=None*)

get_bucket (*bucket_name*, *validate=True*, *headers=None*)

Retrieves a bucket by name.

If the bucket does not exist, an `S3ResponseError` will be raised. If you are unsure if the bucket exists or not, you can use the `S3Connection.lookup` method, which will either return a valid bucket or `None`.

If `validate=False` is passed, no request is made to the service (no charge/communication delay). This is only safe to do if you are **sure** the bucket exists.

If the default `validate=True` is passed, a request is made to the service to ensure the bucket exists. Prior to Boto v2.25.0, this fetched a list of keys (but with a max limit set to 0, always returning an empty list) in the bucket (& included better error messages), at an increased expense. As of Boto v2.25.0, this now performs a HEAD request (less expensive but worse error messages).

If you were relying on parsing the error message before, you should call something like:

```
bucket = conn.get_bucket('<bucket_name>', validate=False)
bucket.get_all_keys(maxkeys=0)
```

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **validate** (*boolean*) – If `True`, it will try to verify the bucket exists on the service-side. (Default: `True`)

get_canonical_user_id (*headers=None*)

Convenience method that returns the “CanonicalUserID” of the user who’s credentials are associated with the connection. The only way to get this value is to do a GET request on the service which returns all buckets associated with the account. As part of that response, the canonical userid is returned. This method simply does all of that and then returns just the user id.

Return type string

Returns A string containing the canonical user id.

head_bucket (*bucket_name*, *headers=None*)

Determines if a bucket exists by name.

If the bucket does not exist, an `S3ResponseError` will be raised.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.

Returns A `<Bucket>` object

lookup (*bucket_name*, *validate=True*, *headers=None*)

Attempts to get a bucket from S3.

Works identically to `S3Connection.get_bucket`, save for that it will return `None` if the bucket does not exist instead of throwing an exception.

Parameters

- **bucket_name** (*string*) – The name of the bucket
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **validate** (*boolean*) – If `True`, it will try to fetch all keys within the given bucket. (Default: `True`)

make_request (*method*, *bucket*=", *key*=", *headers*=None, *data*=", *query_args*=None, *sender*=None, *override_num_retries*=None, *retry_handler*=None)
Makes a request to the server, with stock multiple-retry logic.

set_bucket_class (*bucket_class*)
Set the Bucket class associated with this bucket. By default, this would be the boto.s3.key.Bucket class but if you want to subclass that for some reason this allows you to associate your new class.

Parameters **bucket_class** (*class*) – A subclass of Bucket that can be more specific

class boto.s3.connection.SubdomainCallingFormat

get_bucket_server (***kwargs*)

class boto.s3.connection.VHostCallingFormat

get_bucket_server (***kwargs*)

boto.s3.connection.assert_case_insensitive (*f*)

boto.s3.connection.check_lowercase_bucketname (*n*)

Bucket names must not contain uppercase characters. We check for this by appending a lowercase character and testing with islower(). Note this also covers cases like numeric bucket names with dashes.

```
>>> check_lowercase_bucketname("Aaaa")
Traceback (most recent call last):
...
BotoClientError: S3Error: Bucket names cannot contain upper-case
characters when using either the sub-domain or virtual hosting calling
format.
```

```
>>> check_lowercase_bucketname("1234-5678-9123")
True
>>> check_lowercase_bucketname("abcdefgh1234")
True
```

boto.s3.cors

class boto.s3.cors.CORSConfiguration

A container for the rules associated with a CORS configuration.

add_rule (*allowed_method*, *allowed_origin*, *id*=None, *allowed_header*=None, *max_age_seconds*=None, *expose_header*=None)

Add a rule to this CORS configuration. This only adds the rule to the local copy. To install the new rule(s) on the bucket, you need to pass this CORS config object to the set_cors method of the Bucket object.

Parameters

- **allowed_methods** (*list of str*) – An HTTP method that you want to allow the origin to execute. Each CORSRule must identify at least one origin and one method. Valid values are: GET|PUT|HEAD|POST|DELETE
- **allowed_origin** (*list of str*) – An origin that you want to allow cross-domain requests from. This can contain at most one * wild character. Each CORSRule must identify at least one origin and one method. The origin value can include at most one '*' wild character. For example, "http://*.example.com". You can also specify only * as the origin value allowing all origins cross-domain access.

- **id** (*str*) – A unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.
- **allowed_header** (*list of str*) – Specifies which headers are allowed in a pre-flight OPTIONS request via the Access-Control-Request-Headers header. Each header name specified in the Access-Control-Request-Headers header must have a corresponding entry in the rule. Amazon S3 will send only the allowed headers in a response that were requested. This can contain at most one * wild character.
- **max_age_seconds** (*int*) – The time in seconds that your browser is to cache the preflight response for the specified resource.
- **expose_header** (*list of str*) – One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object). You add one ExposeHeader element in the rule for each header.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

Returns a string containing the XML version of the Lifecycle configuration as defined by S3.

```
class boto.s3.cors.CORSRule (allowed_method=None, allowed_origin=None, id=None,  

                             allowed_header=None, max_age_seconds=None, ex-  

                             pose_header=None)
```

CORS rule for a bucket.

Variables

- **id** – A unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.
- **allowed_methods** – An HTTP method that you want to allow the origin to execute. Each CORSRule must identify at least one origin and one method. Valid values are: GET|PUT|HEAD|POST|DELETE
- **allowed_origin** – An origin that you want to allow cross-domain requests from. This can contain at most one * wild character. Each CORSRule must identify at least one origin and one method. The origin value can include at most one '*' wild character. For example, "http://*.example.com". You can also specify only * as the origin value allowing all origins cross-domain access.
- **allowed_header** – Specifies which headers are allowed in a pre-flight OPTIONS request via the Access-Control-Request-Headers header. Each header name specified in the Access-Control-Request-Headers header must have a corresponding entry in the rule. Amazon S3 will send only the allowed headers in a response that were requested. This can contain at most one * wild character.
- **max_age_seconds** – The time in seconds that your browser is to cache the preflight response for the specified resource.
- **expose_header** – One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object). You add one ExposeHeader element in the rule for each header.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

boto.s3.deletemarker

class boto.s3.deletemarker.DeleteMarker (*bucket=None, name=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.s3.key

class boto.s3.key.Key (*bucket=None, name=None*)

Represents a key (object) in an S3 bucket.

Variables

- **bucket** – The parent *boto.s3.bucket.Bucket*.
- **name** – The name of this Key object.
- **metadata** – A dictionary containing user metadata that you wish to store with the object or that has been retrieved from an existing object.
- **cache_control** – The value of the *Cache-Control* HTTP header.
- **content_type** – The value of the *Content-Type* HTTP header.
- **content_encoding** – The value of the *Content-Encoding* HTTP header.
- **content_disposition** – The value of the *Content-Disposition* HTTP header.
- **content_language** – The value of the *Content-Language* HTTP header.
- **etag** – The *etag* associated with this object.
- **last_modified** – The string timestamp representing the last time this object was modified in S3.
- **owner** – The ID of the owner of this object.
- **storage_class** – The storage class of the object. Currently, one of: STANDARD | REDUCED_REDUNDANCY | GLACIER
- **md5** – The MD5 hash of the contents of the object.
- **size** – The size, in bytes, of the object.
- **version_id** – The version ID of this object, if it is a versioned object.
- **encrypted** – Whether the object is encrypted while at rest on the server.

BufferSize = 8192

DefaultContentType = 'application/octet-stream'

RestoreBody = '<?xml version="1.0" encoding="UTF-8"?>\n <RestoreRequest xmlns="http://'

add_email_grant (*permission, email_address, headers=None*)

Convenience method that provides a quick way to add an email grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to S3.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ, WRITE, READ_ACP, WRITE_ACP, FULL_CONTROL).

- **email_address** (*string*) – The email address associated with the AWS account you are granting the permission to.
- **recursive** (*boolean*) – A boolean value to controls whether the command will apply the grant to all keys within the bucket or not. The default value is False. By passing a True value, the call will iterate through all keys in the bucket and apply the same grant to each key. CAUTION: If you have a lot of keys, this could take a long time!

add_user_grant (*permission, user_id, headers=None, display_name=None*)

Convenience method that provides a quick way to add a canonical user grant to a key. This method retrieves the current ACL, creates a new grant based on the parameters passed in, adds that grant to the ACL and then PUT's the new ACL back to S3.

Parameters

- **permission** (*string*) – The permission being granted. Should be one of: (READ, WRITE, READ_ACP, WRITE_ACP, FULL_CONTROL).
- **user_id** (*string*) – The canonical user id associated with the AWS account you are granting the permission to.
- **display_name** (*string*) – An option string containing the user's Display Name. Only required on Walrus.

base64md5

base_fields = set(['content-length', 'content-language', 'content-disposition', 'content-encoding', 'content-type'])

base_user_settable_fields = set(['content-disposition', 'content-language', 'content-encoding', 'content-type'])

change_storage_class (*new_storage_class, dst_bucket=None, validate_dst_bucket=True*)

Change the storage class of an existing key. Depending on whether a different destination bucket is supplied or not, this will either move the item within the bucket, preserving all metadata and ACL info bucket changing the storage class or it will copy the item to the provided destination bucket, also preserving metadata and ACL info.

Parameters

- **new_storage_class** (*string*) – The new storage class for the Key. Possible values are: * STANDARD * REDUCED_REDUNDANCY
- **dst_bucket** (*string*) – The name of a destination bucket. If not provided the current bucket of the key will be used.
- **validate_dst_bucket** (*bool*) – If True, will validate the dst_bucket by using an extra list request.

close (*fast=False*)

Close this key.

Parameters **fast** (*bool*) – True if you want the connection to be closed without first reading the content. This should only be used in cases where subsequent calls don't need to return the content from the open HTTP connection. Note: As explained at <http://docs.python.org/2/library/httplib.html#httplib.HTTPConnection.getresponse>, callers must read the whole response before sending a new request to the server. Calling Key.close(fast=True) and making a subsequent request to the server will work because boto will get an httplib exception and close/reopen the connection.

closed = False

compute_md5 (*fp, size=None*)

Parameters

- **fp** (*file*) – File pointer to the file to MD5 hash. The file pointer will be reset to the same position before the method returns.

- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where the file is being split in place into different parts. Less bytes may be available.

copy (*dst_bucket, dst_key, metadata=None, reduced_redundancy=False, preserve_acl=False, encrypt_key=False, validate_dst_bucket=True*)
Copy this Key to another bucket.

Parameters

- **dst_bucket** (*string*) – The name of the destination bucket
- **dst_key** (*string*) – The name of the destination key
- **metadata** (*dict*) – Metadata to be associated with new key. If metadata is supplied, it will replace the metadata of the source key being copied. If no metadata is supplied, the source key's metadata will be copied to the new key.
- **reduced_redundancy** (*bool*) – If True, this will force the storage class of the new Key to be REDUCED_REDUNDANCY regardless of the storage class of the key being copied. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **preserve_acl** (*bool*) – If True, the ACL from the source key will be copied to the destination key. If False, the destination key will have the default ACL. Note that preserving the ACL in the new key object will require two additional API calls to S3, one to retrieve the current ACL and one to set that ACL on the new object. If you don't care about the ACL, a value of False will be significantly more efficient.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **validate_dst_bucket** (*bool*) – If True, will validate the dst_bucket by using an extra list request.

Return type *boto.s3.key.Key* or subclass

Returns An instance of the newly created key object

delete (*headers=None*)
Delete this key from S3

endElement (*name, value, connection*)

exists (*headers=None*)
Returns True if the key exists

Return type *bool*

Returns Whether the key exists on S3

f = 'x-robots-tag'

generate_url (*expires_in, method='GET', headers=None, query_auth=True, force_http=False, response_headers=None, expires_in_absolute=False, version_id=None, policy=None, reduced_redundancy=False, encrypt_key=False*)
Generate a URL to access this key.

Parameters

- **expires_in** (*int*) – How long the url is valid for, in seconds.
- **method** (*string*) – The method to use for retrieving the file (default is GET).
- **headers** (*dict*) – Any headers to pass along in the request.

- **query_auth** (*bool*) – If True, signs the request in the URL.
- **force_http** (*bool*) – If True, http will be used instead of https.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **expires_in_absolute** (*bool*) –
- **version_id** (*string*) – The version_id of the object to GET. If specified this overrides any value in the key.
- **policy** (`boto.s3.acl.CannedACLStrings`) – A canned ACL policy that will be applied to the new key in S3.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.

Return type string

Returns The URL to access the key

get_acl (*headers=None*)

get_contents_as_string (*headers=None, cb=None, num_cb=10, torrent=False, version_id=None, response_headers=None, encoding=None*)

Retrieve an object from S3 using the name of the Key object as the key in S3. Return the contents of the object as a string. See `get_contents_to_file` method for details about the parameters.

Parameters

- **headers** (*dict*) – Any additional headers to send in the request
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.
- **encoding** (*str*) – The text encoding to use, such as `utf-8` or `iso-8859-1`. If set, then a string will be returned. Defaults to `None` and returns bytes.

Return type bytes or str

Returns The contents of the file as bytes or a string

get_contents_to_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *res_download_handler=None*, *response_headers=None*)

Retrieve an object from S3 using the name of the Key object as the key in S3. Write the contents of the object to the file pointed to by 'fp'.

Parameters

- **fp** (*File -like object*) –
- **headers** (*dict*) – additional HTTP headers that will be sent with the GET request.
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the cb parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **res_download_handler** – If provided, this handler will perform the download.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to None to always grab the latest version from a version-enabled bucket.

get_contents_to_filename (*filename*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *res_download_handler=None*, *response_headers=None*)

Retrieve an object from S3 using the name of the Key object as the key in S3. Store contents of the object to a file named by 'filename'. See `get_contents_to_file` method for details about the parameters.

Parameters

- **filename** (*string*) – The filename of where to put the file contents
- **headers** (*dict*) – Any additional headers to send in the request
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – If True, returns the contents of a torrent file as a string.
- **res_download_handler** – If provided, this handler will perform the download.

- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.

get_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *torrent=False*, *version_id=None*, *override_num_retries=None*, *response_headers=None*)

Retrieves a file from an S3 Key

Parameters

- **fp** (*file*) – File pointer to put the data into
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – Flag for whether to get a torrent for the file
- **override_num_retries** (*int*) – If not `None` will override configured `num_retries` parameter for underlying GET.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.
- **version_id** (*str*) – The ID of a particular version of the object. If this parameter is not supplied but the Key object has a `version_id` attribute, that value will be used when retrieving the object. You can set the Key object's `version_id` attribute to `None` to always grab the latest version from a version-enabled bucket.

Param headers to send when retrieving the files

get_md5_from_hexdigest (*md5_hexdigest*)

A utility function to create the 2-tuple (`md5hexdigest`, `base64md5`) from just having a precalculated `md5_hexdigest`.

get_metadata (*name*)

get_redirect ()

Return the redirect location configured for this key.

If no redirect is configured (via `set_redirect`), then `None` will be returned.

get_torrent_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*)

Get a torrent file (see to `get_file`)

Parameters

- **fp** (*file*) – The file pointer of where to put the torrent
- **headers** (*dict*) – Headers to be passed

- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the **cb** parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.

get_xml_acl (*headers=None*)

handle_addl_headers (*headers*)

Used by Key subclasses to do additional, provider-specific processing of response headers. No-op for this base class.

handle_encryption_headers (*resp*)

handle_restore_headers (*response*)

handle_storage_class_header (*resp*)

handle_version_headers (*resp, force=False*)

key

make_public (*headers=None*)

md5

next ()

By providing a next method, the key object supports use as an iterator. For example, you can now say:

for bytes in key: write bytes to a file or whatever

All of the HTTP connection stuff is handled for you.

open (*mode='r', headers=None, query_args=None, override_num_retries=None*)

open_read (*headers=None, query_args="", override_num_retries=None, response_headers=None*)

Open this key for reading

Parameters

- **headers** (*dict*) – Headers to pass in the web request
- **query_args** (*string*) – Arguments to pass in the query string (ie, 'torrent')
- **override_num_retries** (*int*) – If not None will override configured **num_retries** parameter for underlying GET.
- **response_headers** (*dict*) – A dictionary containing HTTP headers/values that will override any headers associated with the stored object in the response. See <http://goo.gl/EWOPb> for details.

open_write (*headers=None, override_num_retries=None*)

Open this key for writing. Not yet implemented

Parameters

- **headers** (*dict*) – Headers to pass in the write request
- **override_num_retries** (*int*) – If not None will override configured **num_retries** parameter for underlying PUT.

provider

read (*size=0*)

restore (*days*, *headers=None*)

Restore an object from an archive.

Parameters *days* (*int*) – The lifetime of the restored object (must be at least 1 day). If the object is already restored then this parameter can be used to readjust the lifetime of the restored object. In this case, the *days* param is with respect to the initial time of the request. If the object has not been restored, this param is with respect to the completion time of the request.

send_file (*fp*, *headers=None*, *cb=None*, *num_cb=10*, *query_args=None*, *chunked_transfer=False*, *size=None*)

Upload a file to a key into a bucket on S3.

Parameters

- **fp** (*file*) – The file pointer to upload. The file pointer must point at the offset from which you wish to upload. ie. if uploading the full file, it should point at the start of the file. Normally when a file is opened for reading, the *fp* will point at the first byte. See the *bytes* parameter below for more info.
- **headers** (*dict*) – The headers to pass along with the PUT request
- **num_cb** (*int*) – (optional) If a callback is specified with the *cb* parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer. Providing a negative integer will cause your callback to be called with each buffer read.
- **query_args** (*string*) – (optional) Arguments to pass in the query string.
- **chunked_transfer** (*boolean*) – (optional) If true, we use chunked Transfer-Encoding.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (*fp*). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the default behaviour is to read all bytes from the file pointer. Less bytes may be available.

set_acl (*acl_str*, *headers=None*)

set_canned_acl (*acl_str*, *headers=None*)

set_contents_from_file (*fp*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*, *md5=None*, *reduced_redundancy=False*, *query_args=None*, *encrypt_key=False*, *size=None*, *rewind=False*)

Store an object in S3 using the name of the Key object as the key in S3 and the contents of the file pointed to by ‘*fp*’ as the contents. The data is read from ‘*fp*’ from its current position until ‘*size*’ bytes have been read or EOF.

Parameters

- **fp** (*file*) – the file whose contents to upload
- **headers** (*dict*) – Additional HTTP headers that will be sent with the PUT request.
- **replace** (*bool*) – If this parameter is False, the method will first check to see if an object exists in the bucket with the same key. If it does, it won’t overwrite it. The default value is True which will overwrite the object.
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.

- **num_cb** (*int*) – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (`boto.s3.acl.CannedACLStrings`) – A canned ACL policy that will be applied to the new key in S3.
- **md5** (*A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.*) – If you need to compute the MD5 for any reason prior to upload, it's silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (`fp`). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the default behaviour is to read all bytes from the file pointer. Less bytes may be available.
- **rewind** (*bool*) – (optional) If True, the file pointer (`fp`) will be rewound to the start before any bytes are read from it. The default behaviour is False which reads from the current position of the file pointer (`fp`).

Return type `int`

Returns The number of bytes written to the key.

set_contents_from_filename (*filename, headers=None, replace=True, cb=None, num_cb=10, policy=None, md5=None, reduced_redundancy=False, encrypt_key=False*)

Store an object in S3 using the name of the Key object as the key in S3 and the contents of the file named by 'filename'. See `set_contents_from_file` method for details about the parameters.

Parameters

- **filename** (*string*) – The name of the file that you want to put onto S3
- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **replace** (*bool*) – If True, replaces the contents of the file if it already exists.
- **cb** (*int*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** – (optional) If a callback is specified with the `cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (`boto.s3.acl.CannedACLStrings`) – A canned ACL policy that will be applied to the new key in S3.

- **md5** (A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.) – If you need to compute the MD5 for any reason prior to upload, it’s silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost. :type encrypt_key: bool :param encrypt_key: If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.

Return type `int`

Returns The number of bytes written to the key.

set_contents_from_stream (*fp*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*, *reduced_redundancy=False*, *query_args=None*, *size=None*)

Store an object using the name of the Key object as the key in cloud and the contents of the data stream pointed to by ‘fp’ as the contents.

The stream object is not seekable and total size is not known. This has the implication that we can’t specify the Content-Size and Content-MD5 in the header. So for huge uploads, the delay in calculating MD5 is avoided but with a penalty of inability to verify the integrity of the uploaded data.

Parameters

- **fp** (*file*) – the file whose contents are to be uploaded
- **headers** (*dict*) – additional HTTP headers to be sent with the PUT request.
- **replace** (*bool*) – If this parameter is False, the method will first check to see if an object exists in the bucket with the same key. If it does, it won’t overwrite it. The default value is True which will overwrite the object.
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to GS and the second representing the total number of bytes that need to be transmitted.
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter, this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (*boto.gs.acl.CannedACLStrings*) – A canned ACL policy that will be applied to the new key in GS.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **size** (*int*) – (optional) The Maximum number of bytes to read from the file pointer (fp). This is useful when uploading a file in multiple parts where you are splitting the file up into different ranges to be uploaded. If not specified, the default behaviour is to read all bytes from the file pointer. Less bytes may be available.

set_contents_from_string (*string_data*, *headers=None*, *replace=True*, *cb=None*, *num_cb=10*, *policy=None*, *md5=None*, *reduced_redundancy=False*, *encrypt_key=False*)

Store an object in S3 using the name of the Key object as the key in S3 and the string 's' as the contents. See `set_contents_from_file` method for details about the parameters.

Parameters

- **headers** (*dict*) – Additional headers to pass along with the request to AWS.
- **replace** (*bool*) – If True, replaces the contents of the file if it already exists.
- **cb** (*function*) – a callback function that will be called to report progress on the upload. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted to S3 and the second representing the size of the to be transmitted object.
- **num_cb** (*int*) – (optional) If a callback is specified with the `num_cb` parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **policy** (`boto.s3.acl.CannedACLStrings`) – A canned ACL policy that will be applied to the new key in S3.
- **md5** (A tuple containing the hexdigest version of the MD5 checksum of the file as the first element and the Base64-encoded version of the plain checksum as the second element. This is the same format returned by the `compute_md5` method.) – If you need to compute the MD5 for any reason prior to upload, it's silly to have to do it twice so this param, if present, will be used as the MD5 values of the file. Otherwise, the checksum will be computed.
- **reduced_redundancy** (*bool*) – If True, this will set the storage class of the new Key to be REDUCED_REDUNDANCY. The Reduced Redundancy Storage (RRS) feature of S3, provides lower redundancy at lower storage cost.
- **encrypt_key** (*bool*) – If True, the new copy of the object will be encrypted on the server-side by S3 and will be stored in an encrypted form while at rest in S3.

set_metadata (*name*, *value*)

set_redirect (*redirect_location*, *headers=None*)

Configure this key to redirect to another location.

When the bucket associated with this key is accessed from the website endpoint, a 301 redirect will be issued to the specified *redirect_location*.

Parameters **redirect_location** (*string*) – The location to redirect.

set_remote_metadata (*metadata_plus*, *metadata_minus*, *preserve_acl*, *headers=None*)

set_xml_acl (*acl_str*, *headers=None*)

should_retry (*response*, *chunked_transfer=False*)

startElement (*name*, *attrs*, *connection*)

storage_class

update_metadata (*d*)

boto.s3.prefix

class boto.s3.prefix.**Prefix** (*bucket=None, name=None*)

endElement (*name, value, connection*)

provider

startElement (*name, attrs, connection*)

boto.s3.multipart

class boto.s3.multipart.**CompleteMultiPartUpload** (*bucket=None*)

Represents a completed MultiPart Upload. Contains the following useful attributes:

- **location** - The URI of the completed upload
- **bucket_name** - The name of the bucket in which the upload is contained
- **key_name** - The name of the new, completed key
- **etag** - The MD5 hash of the completed, combined upload
- **version_id** - The version_id of the completed upload
- **encrypted** - The value of the encryption header

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.s3.multipart.**MultiPartUpload** (*bucket=None*)

Represents a MultiPart Upload operation.

cancel_upload ()

Cancels a MultiPart Upload operation. The storage consumed by any previously uploaded parts will be freed. However, if any part uploads are currently in progress, those part uploads might or might not succeed. As a result, it might be necessary to abort a given multipart upload multiple times in order to completely free all storage consumed by all parts.

complete_upload ()

Complete the MultiPart Upload operation. This method should be called when all parts of the file have been successfully uploaded to S3.

Return type boto.s3.multipart.CompletedMultiPartUpload

Returns An object representing the completed upload.

copy_part_from_key (*src_bucket_name, src_key_name, part_num, start=None, end=None, src_version_id=None, headers=None*)

Copy another part of this MultiPart Upload.

Parameters

- **src_bucket_name** (*string*) – Name of the bucket containing the source key
- **src_key_name** (*string*) – Name of the source key
- **part_num** (*int*) – The number of this part.
- **start** (*int*) – Zero-based byte offset to start copying from
- **end** (*int*) – Zero-based byte offset to copy to
- **src_version_id** (*string*) – version_id of source object to copy from
- **headers** (*dict*) – Any headers to pass along in the request

endElement (*name, value, connection*)

get_all_parts (*max_parts=None, part_number_marker=None, encoding_type=None*)

Return the uploaded parts of this MultiPart Upload. This is a lower-level method that requires you to manually page through results. To simplify this process, you can just use the object itself as an iterator and it will automatically handle all of the paging with S3.

startElement (*name, attrs, connection*)

to_xml ()

upload_part_from_file (*fp, part_num, headers=None, replace=True, cb=None, num_cb=10, md5=None, size=None*)

Upload another part of this MultiPart Upload.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

Parameters

- **fp** (*file*) – The file object you want to upload.
- **part_num** (*int*) – The number of this part.

The other parameters are exactly as defined for the *boto.s3.key.Key* `set_contents_from_file` method.

Return type *boto.s3.key.Key* or subclass

Returns The uploaded part containing the etag.

class `boto.s3.multipart.Part` (*bucket=None*)

Represents a single part in a MultiPart upload. Attributes include:

- **part_number** - The integer part number
- **last_modified** - The last modified date of this part
- **etag** - The MD5 hash of this part
- **size** - The size, in bytes, of this part

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

`boto.s3.multipart.part_lister` (*mpupload, part_number_marker=None*)

A generator function for listing parts of a multipart upload.

boto.s3.multidelete

class `boto.s3.multidelete.Deleted` (*key=None, version_id=None, delete_marker=False, delete_marker_version_id=None*)

A successfully deleted object in a multi-object delete request.

Variables

- **key** – Key name of the object that was deleted.
- **version_id** – Version id of the object that was deleted.
- **delete_marker** – If True, indicates the object deleted was a DeleteMarker.
- **delete_marker_version_id** – Version ID of the delete marker deleted.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.s3.multidelete.**Error** (*key=None, version_id=None, code=None, message=None*)
 An unsuccessful deleted object in a multi-object delete request.

Variables

- **key** – Key name of the object that was not deleted.
- **version_id** – Version id of the object that was not deleted.
- **code** – Status code of the failed delete operation.
- **message** – Status message of the failed delete operation.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.s3.multidelete.**MultiDeleteResult** (*bucket=None*)

The status returned from a MultiObject Delete request.

Variables

- **deleted** – A list of successfully deleted objects. Note that if the quiet flag was specified in the request, this list will be empty because only error responses would be returned.
- **errors** – A list of unsuccessfully deleted objects.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.s3.resumable_download_handler

class boto.s3.resumable_download_handler.**ByteTranslatingCallbackHandler** (*proxied_cb, down-load_start_point*)

Proxy class that translates progress callbacks made by boto.s3.Key.get_file(), taking into account that we're resuming a download.

call (*total_bytes_uploaded, total_size*)

class boto.s3.resumable_download_handler.**ResumableDownloadHandler** (*tracker_file_name=None, num_retries=None*)

Handler for resumable downloads.

Constructor. Instantiate once for each downloaded file.

Parameters

- **tracker_file_name** (*string*) – optional file name to save tracking info about this download. If supplied and the current process fails the download, it can be retried in a new process. If called with an existing file containing an unexpired timestamp, we'll resume the transfer for this file; else we'll start a new resumable download.
- **num_retries** (*int*) – the number of times we'll re-try a resumable download making no progress. (Count resets every time we get progress, so download can span many more than this number of retries.)

MIN_ETAG_LEN = 5

RETRYABLE_EXCEPTIONS = (<class 'httplib.HTTPException'>, <type 'exceptions IOError'>)

get_file (*key, fp, headers, cb=None, num_cb=10, torrent=False, version_id=None, hash_algs=None*)

Retrieves a file from a Key :type key: *boto.s3.key.Key* or subclass :param key: The Key object from which upload is to be downloaded

Parameters

- **fp** (*file*) – File pointer into which data should be downloaded
- **cb** (*function*) – (optional) a callback function that will be called to report progress on the download. The callback should accept two integer parameters, the first representing the number of bytes that have been successfully transmitted from the storage service and the second representing the total number of bytes that need to be transmitted.
- **num_cb** (*int*) – (optional) If a callback is specified with the cb parameter this parameter determines the granularity of the callback by defining the maximum number of times the callback will be called during the file transfer.
- **torrent** (*bool*) – Flag for whether to get a torrent for the file
- **version_id** (*string*) – The version ID (optional)
- **hash_algs** (*dictionary*) – (optional) Dictionary of hash algorithms and corresponding hashing class that implements update() and digest(). Defaults to {'md5': hashlib.md5}.

Param headers to send when retrieving the files

Raises ResumableDownloadException if a problem occurs during the transfer.

`boto.s3.resumable_download_handler.get_cur_file_size(fp, position_to_eof=False)`
Returns size of file, optionally leaving fp positioned at EOF.

boto.s3.lifecycle

class `boto.s3.lifecycle.Expiration` (*days=None, date=None*)

When an object will expire.

Variables

- **days** – The number of days until the object expires
- **date** – The date when the object will expire. Must be in ISO 8601 format.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class `boto.s3.lifecycle.Lifecycle`

A container for the rules associated with a Lifecycle configuration.

add_rule (*id=None, prefix="", status='Enabled', expiration=None, transition=None*)

Add a rule to this Lifecycle configuration. This only adds the rule to the local copy. To install the new rule(s) on the bucket, you need to pass this Lifecycle config object to the `configure_lifecycle` method of the Bucket object.

Parameters

- **id** (*str*) – Unique identifier for the rule. The value cannot be longer than 255 characters. This value is optional. The server will generate a unique value for the rule if no value is provided.
- **status** (*str*) – If 'Enabled', the rule is currently being applied. If 'Disabled', the rule is not currently being applied.
- **expiration** (*int*) – Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer. A Expiration object instance is also perfect.

- **transition** (*Transitions*) – Indicates when an object transitions to a different storage class.

Iparam prefix Prefix identifying one or more objects to which the rule applies.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

Returns a string containing the XML version of the Lifecycle configuration as defined by S3.

class `boto.s3.lifecycle.Rule` (*id=None, prefix=None, status=None, expiration=None, transition=None*)

A Lifecycle rule for an S3 bucket.

Variables

- **id** – Unique identifier for the rule. The value cannot be longer than 255 characters. This value is optional. The server will generate a unique value for the rule if no value is provided.
- **prefix** – Prefix identifying one or more objects to which the rule applies. If prefix is not provided, Boto generates a default prefix which will match all objects.
- **status** – If ‘Enabled’, the rule is currently being applied. If ‘Disabled’, the rule is not currently being applied.
- **expiration** – An instance of *Expiration*. This indicates the lifetime of the objects that are subject to the rule.
- **transition** – An instance of *Transition*. This indicates when to transition to a different storage class.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class `boto.s3.lifecycle.Transition` (*days=None, date=None, storage_class=None*)

A transition to a different storage class.

Variables

- **days** – The number of days until the object should be moved.
- **date** – The date when the object should be moved. Should be in ISO 8601 format.
- **storage_class** – The storage class to transition to. Valid values are GLACIER, STANDARD_IA.

to_xml ()

class `boto.s3.lifecycle.Transitions`

A container for the transitions associated with a Lifecycle’s Rule configuration.

add_transition (*days=None, date=None, storage_class=None*)

Add a transition to this Lifecycle configuration. This only adds the rule to the local copy. To install the new rule(s) on the bucket, you need to pass this Lifecycle config object to the `configure_lifecycle` method of the Bucket object.

Variables

- **days** – The number of days until the object should be moved.
- **date** – The date when the object should be moved. Should be in ISO 8601 format.

- *storage_class* – The storage class to transition to. Valid values are GLACIER, STANDARD_IA.

date

days

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

storage_class

to_xml ()

Returns a string containing the XML version of the Lifecycle configuration as defined by S3.

boto.s3.tagging

class boto.s3.tagging.**Tag** (*key=None, value=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class boto.s3.tagging.**TagSet**

add_tag (*key, value*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

class boto.s3.tagging.**Tags**

A container for the tags associated with a bucket.

add_tag_set (*tag_set*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml ()

boto.s3.user

class boto.s3.user.**User** (*parent=None, id="", display_name=""*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

to_xml (*element_name='Owner'*)

4.138.10 SDB Reference

In addition to what is seen below, boto includes an abstraction layer for SimpleDB that may be used:

- *SimpleDB DB* (Maintained, but little documentation)

boto.sdb

`boto.sdb.connect_to_region(region_name, **kw_params)`

Given a valid region name, return a `boto.sdb.connection.SDBConnection`.

Type `str`

Parameters `region_name` – The name of the region to connect to.

Return type `boto.sdb.connection.SDBConnection` or `None`

Returns A connection to the given region, or `None` if an invalid region name is given

`boto.sdb.regions()`

Get all available regions for the SDB service.

Return type `list`

Returns A list of `boto.sdb.regioninfo.RegionInfo` instances

boto.sdb.connection

class `boto.sdb.connection.ItemThread(name, domain_name, item_names)`

A threaded *Item* retriever utility class. Retrieved *Item* objects are stored in the `items` instance variable after `run()` is called.

Tip: The item retrieval will not start until the `run()` method is called.

Parameters

- **name** (*str*) – A thread name. Used for identification.
- **domain_name** (*str*) – The name of a SimpleDB *Domain*
- **item_names** (*string or list of strings*) – The name(s) of the items to retrieve from the specified *Domain*.

Variables `items` (*list*) – A list of items retrieved. Starts as empty list.

run()

Start the threaded retrieval of items. Populates the `items` list with *Item* objects.

class `boto.sdb.connection.SDBConnection(aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, proxy_user=None, proxy_pass=None, debug=0, https_connection_factory=None, region=None, path='/', converter=None, security_token=None, validate_certs=True, profile_name=None)`

This class serves as a gateway to your SimpleDB region (defaults to us-east-1). Methods within allow access to SimpleDB *Domain* objects and their associated *Item* objects.

Tip: While you may instantiate this class directly, it may be easier to go through `boto.connect_sdb()`.

For any keywords that aren't documented, refer to the parent class, `boto.connection.AWSAuthConnection`. You can avoid having to worry about these keyword arguments by instantiating these objects via `boto.connect_sdb()`.

Parameters `region` (`boto.sdb.regioninfo.SDBRegionInfo`) – Explicitly specify a region. Defaults to `us-east-1` if not specified. You may also specify the region in your `boto.cfg`:

```
[SDB]
region = eu-west-1
```

```
APIVersion = '2009-04-15'
```

```
DefaultRegionEndpoint = 'sdb.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

ResponseError

alias of `boto.exception.SDBResponseError`

batch_delete_attributes (*domain_or_name, items*)

Delete multiple items in a domain.

Parameters

- **domain_or_name** (string or `boto.sdb.domain.Domain` object.) – Either the name of a domain or a `Domain` object
- **items** (*dict or dict-like object*) – A dictionary-like object. The keys of the dictionary are the item names and the values are either:
 - dictionaries of attribute names/values, exactly the same as the `attribute_names` parameter of the scalar `put_attributes` call. The attribute name/value pairs will only be deleted if they match the name/value pairs passed in.
 - None which means that all attributes associated with the item should be deleted.

Returns True if successful

batch_put_attributes (*domain_or_name, items, replace=True*)

Store attributes for multiple items in a domain.

Parameters

- **domain_or_name** (string or `boto.sdb.domain.Domain` object.) – Either the name of a domain or a `Domain` object
- **items** (*dict or dict-like object*) – A dictionary-like object. The keys of the dictionary are the item names and the values are themselves dictionaries of attribute names/values, exactly the same as the `attribute_names` parameter of the scalar `put_attributes` call.
- **replace** (*bool*) – Whether the attribute values passed in will replace existing values or will be added as addition values. Defaults to True.

Return type `bool`

Returns True if successful

create_domain (*domain_name*)

Create a SimpleDB domain.

Parameters `domain_name` (*string*) – The name of the new domain

Return type `boto.sdb.domain.Domain` object

Returns The newly created domain

delete_attributes (*domain_or_name, item_name, attr_names=None, expected_value=None*)

Delete attributes from a given item in a domain.

Parameters

- **domain_or_name** (string or *boto.sdb.domain.Domain* object.) – Either the name of a domain or a Domain object
- **item_name** (*string*) – The name of the item whose attributes are being deleted.
- **attributes** (dict, list or *boto.sdb.item.Item*) – Either a list containing attribute names which will cause all values associated with that attribute name to be deleted or a dict or Item containing the attribute names and keys and list of values to delete as the value. If no value is supplied, all attribute name/values for the item will be deleted.
- **expected_value** (*list*) – If supplied, this is a list or tuple consisting of a single attribute name and expected value. The list can be of the form:

– ['name', 'value']

In which case the call will first verify that the attribute “name” of this item has a value of “value”. If it does, the delete will proceed, otherwise a ConditionalCheckFailed error will be returned. The list can also be of the form:

– ['name', True|False]

which will simply check for the existence (True) or non-existence (False) of the attribute.

Return type *bool*

Returns True if successful

delete_domain (*domain_or_name*)

Delete a SimpleDB domain.

Caution: This will delete the domain and all items within the domain.

Parameters **domain_or_name** (string or *boto.sdb.domain.Domain* object.) – Either the name of a domain or a Domain object

Return type *bool*

Returns True if successful

domain_metadata (*domain_or_name*)

Get the Metadata for a SimpleDB domain.

Parameters **domain_or_name** (string or *boto.sdb.domain.Domain* object.) – Either the name of a domain or a Domain object

Return type *boto.sdb.domain.DomainMetaData* object

Returns The newly created domain metadata object

get_all_domains (*max_domains=None, next_token=None*)

Returns a *boto.resultset.ResultSet* containing all *boto.sdb.domain.Domain* objects associated with this connection’s Access Key ID.

Parameters

- **max_domains** (*int*) – Limit the returned *ResultSet* to the specified number of members.
- **next_token** (*str*) – A token string that was returned in an earlier call to this method as the *next_token* attribute on the returned *ResultSet* object. This attribute is set if there are more than Domains than the value specified in the *max_domains* keyword. Pass the *next_token* value from your earlier query in this keyword to get the next ‘page’ of domains.

get_attributes (*domain_or_name*, *item_name*, *attribute_names=None*, *consistent_read=False*, *item=None*)

Retrieve attributes for a given item in a domain.

Parameters

- **domain_or_name** (string or *boto.sdb.domain.Domain* object.) – Either the name of a domain or a Domain object
- **item_name** (*string*) – The name of the item whose attributes are being retrieved.
- **attribute_names** (*string* or *list of strings*) – An attribute name or list of attribute names. This parameter is optional. If not supplied, all attributes will be retrieved for the item.
- **consistent_read** (*bool*) – When set to true, ensures that the most recent data is returned.
- **item** (*boto.sdb.item.Item*) – Instead of instantiating a new Item object, you may specify one to update.

Return type *boto.sdb.item.Item*

Returns An Item with the requested attribute name/values set on it

get_domain (*domain_name*, *validate=True*)

Retrieves a *boto.sdb.domain.Domain* object whose name matches *domain_name*.

Parameters

- **domain_name** (*str*) – The name of the domain to retrieve
- **validate** (*bool*) – When True, check to see if the domain actually exists. If False, blindly return a *Domain* object with the specified name set.

Raises *boto.exception.SDBResponseError* if *validate* is True and no match could be found.

Return type *boto.sdb.domain.Domain*

Returns The requested domain

get_domain_and_name (*domain_or_name*)

Given a *str* or *boto.sdb.domain.Domain*, return a tuple with the following members (in order):

- In instance of *boto.sdb.domain.Domain* for the requested domain
- The domain’s name as a *str*

Parameters **domain_or_name** (*str* or *boto.sdb.domain.Domain*) – The domain or domain name to get the domain and name for.

Raises *boto.exception.SDBResponseError* when an invalid domain name is specified.

Return type *tuple*

Returns A tuple with contents outlined as per above.

get_usage()

Returns the BoxUsage (in USD) accumulated on this specific SDBConnection instance.

Tip: This can be out of date, and should only be treated as a rough estimate. Also note that this estimate only applies to the requests made on this specific connection instance. It is by no means an account-wide estimate.

Return type `float`

Returns The accumulated BoxUsage of all requests made on the connection.

lookup (*domain_name*, *validate=True*)

Lookup an existing SimpleDB domain. This differs from `get_domain()` in that `None` is returned if `validate` is `True` and no match was found (instead of raising an exception).

Parameters

- **domain_name** (*str*) – The name of the domain to retrieve
- **validate** (*bool*) – If `True`, a `None` value will be returned if the specified domain can't be found. If `False`, a `Domain` object will be dumbly returned, regardless of whether it actually exists.

Return type `boto.sdb.domain.Domain` object or `None`

Returns The Domain object or `None` if the domain does not exist.

print_usage()

Print the BoxUsage and approximate costs of all requests made on this specific SDBConnection instance.

Tip: This can be out of date, and should only be treated as a rough estimate. Also note that this estimate only applies to the requests made on this specific connection instance. It is by no means an account-wide estimate.

put_attributes (*domain_or_name*, *item_name*, *attributes*, *replace=True*, *expected_value=None*)

Store attributes for a given item in a domain.

Parameters

- **domain_or_name** (string or `boto.sdb.domain.Domain` object.) – Either the name of a domain or a Domain object
- **item_name** (*string*) – The name of the item whose attributes are being stored.
- **attribute_names** (*dict* or *dict-like object*) – The name/value pairs to store as attributes
- **expected_value** (*list*) – If supplied, this is a list or tuple consisting of a single attribute name and expected value. The list can be of the form:

– ['name', 'value']

In which case the call will first verify that the attribute “name” of this item has a value of “value”. If it does, the delete will proceed, otherwise a `ConditionalCheckFailed` error will be returned. The list can also be of the form:

– ['name', True/False]

which will simply check for the existence (`True`) or non-existence (`False`) of the attribute.

- **replace** (*bool*) – Whether the attribute values passed in will replace existing values or will be added as addition values. Defaults to True.

Return type *bool*

Returns True if successful

select (*domain_or_name, query="", next_token=None, consistent_read=False*)

Returns a set of Attributes for item names within domain_name that match the query. The query must be expressed in using the SELECT style syntax rather than the original SimpleDB query language. Even though the select request does not require a domain object, a domain object must be passed into this method so the Item objects returned can point to the appropriate domain.

Parameters

- **domain_or_name** (string or *boto.sdb.domain.Domain* object) – Either the name of a domain or a Domain object
- **query** (*string*) – The SimpleDB query to be performed.
- **consistent_read** (*bool*) – When set to true, ensures that the most recent data is returned.

Return type *ResultSet*

Returns An iterator containing the results.

set_item_cls (*cls*)

While the default item class is *boto.sdb.item.Item*, this default may be overridden. Use this method to change a connection's item class.

Parameters **cls** (*object*) – The new class to set as this connection's item class. See the default item class for inspiration as to what your replacement should/could look like.

boto.sdb.domain

class *boto.sdb.domain.Domain* (*connection=None, name=None*)

batch_delete_attributes (*items*)

Delete multiple items in this domain.

Parameters **items** (*dict or dict-like object*) – A dictionary-like object. The keys of the dictionary are the item names and the values are either:

- dictionaries of attribute names/values, exactly the same as the attribute_names parameter of the scalar put_attributes call. The attribute name/value pairs will only be deleted if they match the name/value pairs passed in.
- None which means that all attributes associated with the item should be deleted.

Return type *bool*

Returns True if successful

batch_put_attributes (*items, replace=True*)

Store attributes for multiple items.

Parameters

- **items** (*dict or dict-like object*) – A dictionary-like object. The keys of the dictionary are the item names and the values are themselves dictionaries of attribute names/values, exactly the same as the attribute_names parameter of the scalar put_attributes call.

- **replace** (*bool*) – Whether the attribute values passed in will replace existing values or will be added as addition values. Defaults to True.

Return type `bool`

Returns True if successful

delete ()

Delete this domain, and all items under it

delete_attributes (*item_name*, *attributes=None*, *expected_values=None*)

Delete attributes from a given item.

Parameters

- **item_name** (*string*) – The name of the item whose attributes are being deleted.
- **attributes** (*dict*, *list* or *boto.sdb.item.Item*) – Either a list containing attribute names which will cause all values associated with that attribute name to be deleted or a dict or Item containing the attribute names and keys and list of values to delete as the value. If no value is supplied, all attribute name/values for the item will be deleted.
- **expected_value** (*list*) – If supplied, this is a list or tuple consisting of a single attribute name and expected value. The list can be of the form:

– ['name', 'value']

In which case the call will first verify that the attribute “name” of this item has a value of “value”. If it does, the delete will proceed, otherwise a ConditionalCheckFailed error will be returned. The list can also be of the form:

– ['name', True|False]

which will simply check for the existence (True) or non-existence (False) of the attribute.

Return type `bool`

Returns True if successful

delete_item (*item*)

endElement (*name*, *value*, *connection*)

from_xml (*doc*)

Load this domain based on an XML document

get_attributes (*item_name*, *attribute_name=None*, *consistent_read=False*, *item=None*)

Retrieve attributes for a given item.

Parameters

- **item_name** (*string*) – The name of the item whose attributes are being retrieved.
- **attribute_names** (*string* or *list of strings*) – An attribute name or list of attribute names. This parameter is optional. If not supplied, all attributes will be retrieved for the item.

Return type *boto.sdb.item.Item*

Returns An Item mapping type containing the requested attribute name/values

get_item (*item_name*, *consistent_read=False*)

Retrieves an item from the domain, along with all of its attributes.

Parameters

- **item_name** (*string*) – The name of the item to retrieve.
- **consistent_read** (*bool*) – When set to true, ensures that the most recent data is returned.

Return type *boto.sdb.item.Item* or None

Returns The requested item, or None if there was no match found

get_metadata ()

new_item (*item_name*)

put_attributes (*item_name, attributes, replace=True, expected_value=None*)

Store attributes for a given item.

Parameters

- **item_name** (*string*) – The name of the item whose attributes are being stored.
- **attribute_names** (*dict or dict-like object*) – The name/value pairs to store as attributes
- **expected_value** (*list*) – If supplied, this is a list or tuple consisting of a single attribute name and expected value. The list can be of the form:
 - ['name', 'value']

In which case the call will first verify that the attribute “name” of this item has a value of “value”. If it does, the delete will proceed, otherwise a ConditionalCheckFailed error will be returned. The list can also be of the form:

- ['name', True|False]

which will simply check for the existence (True) or non-existence (False) of the attribute.

- **replace** (*bool*) – Whether the attribute values passed in will replace existing values or will be added as addition values. Defaults to True.

Return type *bool*

Returns True if successful

select (*query=", next_token=None, consistent_read=False, max_items=None*)

Returns a set of Attributes for item names within domain_name that match the query. The query must be expressed in using the SELECT style syntax rather than the original SimpleDB query language.

Parameters **query** (*string*) – The SimpleDB query to be performed.

Return type *iter*

Returns An iterator containing the results. This is actually a generator function that will iterate across all search results, not just the first page.

startElement (*name, attrs, connection*)

to_xml (*f=None*)

Get this domain as an XML DOM Document ;param f: Optional File to dump directly to ;type f: File or Stream

Returns File object where the XML has been dumped to

Return type *file*

```
class boto.sdb.domain.DomainDumpParser (domain)
    SAX parser for a domain that has been dumped
```

```
    characters (ch)
```

```
    endElement (name)
```

```
    startElement (name, attrs)
```

```
class boto.sdb.domain.DomainMetaData (domain=None)
```

```
    endElement (name, value, connection)
```

```
    startElement (name, attrs, connection)
```

```
class boto.sdb.domain.UploaderThread (domain)
```

```
    Uploader Thread
```

```
    run ()
```

```
        Method representing the thread's activity.
```

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

boto.sdb.item

```
class boto.sdb.item.Item (domain, name="", active=False)
```

A dict sub-class that serves as an object representation of a SimpleDB item. An item in SDB is similar to a row in a relational database. Items belong to a *Domain*, which is similar to a table in a relational database.

The keys on instances of this object correspond to attributes that are stored on the SDB item.

Tip: While it is possible to instantiate this class directly, you may want to use the convenience methods on `boto.sdb.domain.Domain` for that purpose. For example, `boto.sdb.domain.Domain.get_item()`.

Parameters

- **domain** (`boto.sdb.domain.Domain`) – The domain that this item belongs to.
- **name** (`str`) – The name of this item. This name will be used when querying for items using methods like `boto.sdb.domain.Domain.get_item()`

```
add_value (key, value)
```

Helps set or add to attributes on this item. If you are adding a new attribute that has yet to be set, it will simply create an attribute named `key` with your given `value` as its value. If you are adding a value to an existing attribute, this method will convert the attribute to a list (if it isn't already) and append your new value to said list.

For clarification, consider the following interactive session:

```
>>> item = some_domain.get_item('some_item')
>>> item.has_key('some_attr')
False
>>> item.add_value('some_attr', 1)
>>> item['some_attr']
1
```

(continues on next page)

(continued from previous page)

```
>>> item.add_value('some_attr', 2)
>>> item['some_attr']
[1, 2]
```

Parameters

- **key** (*str*) – The attribute to add a value to.
- **value** (*object*) – The value to set or append to the attribute.

decode_value (*value*)**delete** ()

Deletes this item in SDB.

Note: This local Python object remains in its current state after deletion, this only deletes the remote item in SDB.

endElement (*name, value, connection*)**load** ()

Loads or re-loads this item's attributes from SDB.

Warning: If you have changed attribute values on an Item instance, this method will over-write the values if they are different in SDB. For any local attributes that don't yet exist in SDB, they will be safe.

save (*replace=True*)

Saves this item to SDB.

Parameters **replace** (*bool*) – If True, delete any attributes on the remote SDB item that have a None value on this object.

startElement (*name, attrs, connection*)**boto.sdb.queryresultset**

```
class boto.sdb.queryresultset.QueryResultSet (domain=None, query="",  
                                              max_items=None, attr_names=None)
```

```
class boto.sdb.queryresultset.SelectResultSet (domain=None, query="",  
                                              max_items=None, next_token=None,  
                                              consistent_read=False)
```

next ()

```
boto.sdb.queryresultset.query_lister (domain, query="", max_items=None,  
                                       attr_names=None)
```

```
boto.sdb.queryresultset.select_lister (domain, query="", max_items=None)
```

4.138.11 services

boto.services

boto.services.bs

```
class boto.services.bs.BS
```

```

    Commands = {'batches': 'List all batches stored in current output_domain', 'reset':
Usage = 'usage: %prog [options] config_file command'
do_batches()
do_reset()
do_retrieve()
do_start()
do_status()
do_submit()
main()
print_command_help()

```

boto.services.message

```
class boto.services.message.ServiceMessage (queue=None, body=None, xml_attrs=None)
```

```
    for_key (key, params=None, bucket_name=None)
```

boto.services.result

```
class boto.services.result.ResultProcessor (batch_name, sd, mimetype_files=None)
```

```

LogFileName = 'log.csv'
calculate_stats (msg)
get_results (path, get_file=True, delete_msg=True)
get_results_from_bucket (path)
get_results_from_domain (path, get_file=True)
get_results_from_queue (path, get_file=True, delete_msg=True)
log_message (msg, path)
process_record (record, path, get_file=True)

```

boto.services.service

```
class boto.services.service.Service (config_file=None, mimetype_files=None)
```

```
ProcessingTime = 60  
cleanup ()  
delete_message (message)  
get_file (message)  
main (notify=False)  
process_file (in_file_name, msg)  
put_file (bucket_name, file_path, key_name=None)  
read_message ()  
save_results (results, input_message, output_message)  
shutdown ()  
split_key ()  
write_message (message)
```

boto.services.servicedef

```
class boto.services.servicedef.ServiceDef (config_file, aws_access_key_id=None,  
 aws_secret_access_key=None)
```

```
get (name, default=None)
```

```
get_obj (name)
```

Returns the AWS object associated with a given option.

The heuristics used are a bit lame. If the option name contains the word 'bucket' it is assumed to be an S3 bucket, if the name contains the word 'queue' it is assumed to be an SQS queue and if it contains the word 'domain' it is assumed to be a SimpleDB domain. If the option name specified does not exist in the config file or if the AWS object cannot be retrieved this returns None.

```
getbool (option, default=False)
```

```
getint (option, default=0)
```

```
has_option (option)
```

boto.services.sonofmmm

```
class boto.services.sonofmmm.SonOfMMM (config_file=None)
```

```
process_file (in_file_name, msg)
```

```
queue_files ()
```

```
shutdown ()
```

boto.services.submit

class boto.services.submit.**Submitter** (*sd*)

get_key_name (*fullpath, prefix*)

submit_file (*path, metadata=None, cb=None, num_cb=0, prefix='/'*)

submit_path (*path, tags=None, ignore_dirs=None, cb=None, num_cb=0, status=False, prefix='/'*)

write_message (*key, metadata*)

4.138.12 SES**boto.ses**

boto.ses.connect_to_region (*region_name, **kw_params*)

Given a valid region name, return a *boto.ses.connection.SESConnection*.

Type str

Parameters *region_name* – The name of the region to connect to.

Return type *boto.ses.connection.SESConnection* or None

Returns A connection to the given region, or None if an invalid region name is given

boto.ses.regions ()

Get all available regions for the SES service.

Return type list

Returns A list of *boto.regioninfo.RegionInfo* instances

boto.ses.connection

class boto.ses.connection.**SESConnection** (*aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, proxy_user=None, proxy_pass=None, debug=0, https_connection_factory=None, region=None, path='/', security_token=None, validate_certs=True, profile_name=None*)

APIVersion = '2010-12-01'

DefaultRegionEndpoint = 'email.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.BotoServerError*

delete_identity (*identity*)

Deletes the specified identity (email address or domain) from the list of verified identities.

Parameters *identity* (*string*) – The identity to be deleted.

Return type dict

Returns A DeleteIdentityResponse structure. Note that keys must be unicode strings.

delete_verified_email_address (*email_address*)

Deletes the specified email address from the list of verified addresses.

Parameters `email_address` – The email address to be removed from the list of verified addresses.

Return type `dict`

Returns A `DeleteVerifiedEmailAddressResponse` structure. Note that keys must be unicode strings.

get_identity_dkim_attributes (*identities*)

Get attributes associated with a list of verified identities.

Given a list of verified identities (email addresses and/or domains), returns a structure describing identity notification attributes.

Parameters `identities` (*list*) – A list of verified identities (email addresses and/or domains).

get_identity_verification_attributes (*identities*)

Given a list of identities (email addresses and/or domains), returns the verification status and (for domain identities) the verification token for each identity.

Parameters `identities` (*list of strings or string*) – List of identities.

Return type `dict`

Returns A `GetIdentityVerificationAttributesResponse` structure. Note that keys must be unicode strings.

get_send_quota ()

Fetches the user's current activity limits.

Return type `dict`

Returns A `GetSendQuotaResponse` structure. Note that keys must be unicode strings.

get_send_statistics ()

Fetches the user's sending statistics. The result is a list of data points, representing the last two weeks of sending activity.

Each data point in the list contains statistics for a 15-minute interval.

Return type `dict`

Returns A `GetSendStatisticsResponse` structure. Note that keys must be unicode strings.

list_identities ()

Returns a list containing all of the identities (email addresses and domains) for a specific AWS Account, regardless of verification status.

Return type `dict`

Returns A `ListIdentitiesResponse` structure. Note that keys must be unicode strings.

list_verified_email_addresses ()

Fetch a list of the email addresses that have been verified.

Return type `dict`

Returns A `ListVerifiedEmailAddressesResponse` structure. Note that keys must be unicode strings.

send_email (*source*, *subject*, *body*, *to_addresses*, *cc_addresses=None*, *bcc_addresses=None*, *format='text'*, *reply_addresses=None*, *return_path=None*, *text_body=None*, *html_body=None*)

Composes an email message based on input data, and then immediately queues the message for sending.

Parameters

- **source** (*string*) – The sender's email address.

- **subject** (*string*) – The subject of the message: A short summary of the content, which will appear in the recipient’s inbox.
- **body** (*string*) – The message body.
- **to_addresses** (*list of strings or string*) – The To: field(s) of the message.
- **cc_addresses** (*list of strings or string*) – The CC: field(s) of the message.
- **bcc_addresses** (*list of strings or string*) – The BCC: field(s) of the message.
- **format** (*string*) – The format of the message’s body, must be either “text” or “html”.
- **reply_addresses** (*list of strings or string*) – The reply-to email address(es) for the message. If the recipient replies to the message, each reply-to address will receive the reply.
- **return_path** (*string*) – The email address to which bounce notifications are to be forwarded. If the message cannot be delivered to the recipient, then an error message will be returned from the recipient’s ISP; this message will then be forwarded to the email address specified by the ReturnPath parameter.
- **text_body** (*string*) – The text body to send with this email.
- **html_body** (*string*) – The html body to send with this email.

send_raw_email (*raw_message, source=None, destinations=None*)

Sends an email message, with header and content specified by the client. The SendRawEmail action is useful for sending multipart MIME emails, with attachments or inline content. The raw text of the message must comply with Internet email standards; otherwise, the message cannot be sent.

Parameters

- **source** (*string*) – The sender’s email address. Amazon’s docs say:
If you specify the Source parameter, then bounce notifications and complaints will be sent to this email address. This takes precedence over any Return-Path header that you might include in the raw text of the message.
- **raw_message** (*string*) – The raw text of the message. The client is responsible for ensuring the following:
 - Message must contain a header and a body, separated by a blank line.
 - All required header fields must be present.
 - Each part of a multipart MIME message must be formatted properly.
 - MIME content types must be among those supported by Amazon SES. Refer to the Amazon SES Developer Guide for more details.
 - Content must be base64-encoded, if MIME requires it.
- **destinations** (*list of strings or string*) – A list of destinations for the message.

set_identity_dkim_enabled (*identity, dkim_enabled*)

Enables or disables DKIM signing of email sent from an identity.

- If Easy DKIM signing is enabled for a domain name identity (e.g., `example.com`), then Amazon SES will DKIM-sign all email sent by addresses under that domain name (e.g., `user@example.com`)

- If Easy DKIM signing is enabled for an email address, then Amazon SES will DKIM-sign all email sent by that email address.

For email addresses (e.g., `user@example.com`), you can only enable Easy DKIM signing if the corresponding domain (e.g., `example.com`) has been set up for Easy DKIM using the AWS Console or the `VerifyDomainDkim` action.

Parameters

- **identity** (*string*) – An email address or domain name.
- **dkim_enabled** (*bool*) – Specifies whether or not to enable DKIM signing.

set_identity_feedback_forwarding_enabled (*identity, forwarding_enabled=True*)

Enables or disables SES feedback notification via email. Feedback forwarding may only be disabled when both complaint and bounce topics are set.

Parameters

- **identity** (*string*) – An email address or domain name.
- **forwarding_enabled** (*bool*) – Specifies whether or not to enable feedback forwarding.

set_identity_notification_topic (*identity, notification_type, sns_topic=None*)

Sets an SNS topic to publish bounce or complaint notifications for emails sent with the given identity as the Source. Publishing to topics may only be disabled when feedback forwarding is enabled.

Parameters

- **identity** (*string*) – An email address or domain name.
- **notification_type** (*string*) – The type of feedback notifications that will be published to the specified topic. Valid Values: Bounce | Complaint | Delivery
- **sns_topic** (*string or None*) – The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (Amazon SNS) topic.

verify_domain_dkim (*domain*)

Returns a set of DNS records, or tokens, that must be published in the domain name's DNS to complete the DKIM verification process. These tokens are DNS `CNAME` records that point to DKIM public keys hosted by Amazon SES. To complete the DKIM verification process, these tokens must be published in the domain's DNS. The tokens must remain published in order for Easy DKIM signing to function correctly.

After the tokens are added to the domain's DNS, Amazon SES will be able to DKIM-sign email originating from that domain. To enable or disable Easy DKIM signing for a domain, use the `SetIdentityDkimEnabled` action. For more information about Easy DKIM, go to the [Amazon SES Developer Guide](#).

Parameters **domain** (*string*) – The domain name.

verify_domain_identity (*domain*)

Verifies a domain.

Parameters **domain** (*string*) – The domain to be verified.

Return type `dict`

Returns A `VerifyDomainIdentityResponse` structure. Note that keys must be unicode strings.

verify_email_address (*email_address*)

Verifies an email address. This action causes a confirmation email message to be sent to the specified address.

Parameters **email_address** – The email address to be verified.

Return type `dict`

Returns A VerifyEmailAddressResponse structure. Note that keys must be unicode strings.

verify_email_identity (*email_address*)

Verifies an email address. This action causes a confirmation email message to be sent to the specified address.

Parameters **email_address** – The email address to be verified.

Return type dict

Returns A VerifyEmailIdentityResponse structure. Note that keys must be unicode strings.

4.138.13 SNS

boto.sns

`boto.sns.connect_to_region` (*region_name*, ****kw_params**)

Given a valid region name, return a `boto.sns.connection.SNSConnection`.

Type str

Parameters **region_name** – The name of the region to connect to.

Return type `boto.sns.connection.SNSConnection` or None

Returns A connection to the given region, or None if an invalid region name is given

`boto.sns.regions` ()

Get all available regions for the SNS service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo` instances

```
class boto.sns.SNSConnection (aws_access_key_id=None,          aws_secret_access_key=None,
                             is_secure=True, port=None, proxy=None, proxy_port=None,
                             proxy_user=None, proxy_pass=None, debug=0,
                             https_connection_factory=None, region=None, path='/', security_token=None,
                             validate_certs=True, profile_name=None)
```

Amazon Simple Notification Service Amazon Simple Notification Service (Amazon SNS) is a web service that enables you to build distributed web-enabled applications. Applications can use Amazon SNS to easily push real-time notification messages to interested subscribers over multiple delivery protocols. For more information about this product see '<http://aws.amazon.com/sns>'. For detailed information about Amazon SNS features and their associated API calls, see the '[Amazon SNS Developer Guide](#)'.

We also provide SDKs that enable you to access Amazon SNS from your preferred programming language. The SDKs contain functionality that automatically takes care of tasks such as: cryptographically signing your service requests, retrying requests, and handling error responses. For a list of available SDKs, go to '[Tools for Amazon Web Services](#)'.

APIVersion = '2010-03-31'

DefaultRegionEndpoint = 'sns.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

add_permission (*topic*, *label*, *account_ids*, *actions*)

Adds a statement to a topic's access control policy, granting access for the specified AWS accounts to the specified actions.

Parameters

- **topic** (*string*) – The ARN of the topic.
- **label** (*string*) – A unique identifier for the new policy statement.
- **account_ids** (*list of strings*) – The AWS account ids of the users who will be give access to the specified actions.

- **actions** (*list of strings*) – The actions you want to allow for each of the specified principal(s).

confirm_subscription (*topic, token, authenticate_on_unsubscribe=False*)

Get properties of a Topic

Parameters

- **topic** (*string*) – The ARN of the new topic.
- **token** (*string*) – Short-lived token sent to and endpoint during the Subscribe operation.
- **authenticate_on_unsubscribe** (*bool*) – Optional parameter indicating that you wish to disable unauthenticated unsubscription of the subscription.

create_platform_application (*name=None, platform=None, attributes=None*)

The *CreatePlatformApplication* action creates a platform application object for one of the supported push notification services, such as APNS and GCM, to which devices and mobile apps may register. You must specify PlatformPrincipal and PlatformCredential attributes when using the *CreatePlatformApplication* action. The PlatformPrincipal is received from the notification service. For APNS/APNS_SANDBOX, PlatformPrincipal is “SSL certificate”. For GCM, PlatformPrincipal is not applicable. For ADM, PlatformPrincipal is “client id”. The PlatformCredential is also received from the notification service. For APNS/APNS_SANDBOX, PlatformCredential is “private key”. For GCM, PlatformCredential is “API key”. For ADM, PlatformCredential is “client secret”. The PlatformApplicationArn that is returned when using *CreatePlatformApplication* is then used as an attribute for the *CreatePlatformEndpoint* action. For more information, see [‘Using Amazon SNS Mobile Push Notifications’_](#).

Parameters

- **name** (*string*) – Application names must be made up of only uppercase and lowercase ASCII letters, numbers, underscores, hyphens, and periods, and must be between 1 and 256 characters long.
- **platform** (*string*) – The following platforms are supported: ADM (Amazon Device Messaging), APNS (Apple Push Notification Service), APNS_SANDBOX, and GCM (Google Cloud Messaging).
- **attributes** (*map*) – For a list of attributes, see [‘SetPlatformApplicationAttributes’_](#)

create_platform_endpoint (*platform_application_arn=None, token=None, custom_user_data=None, attributes=None*)

The *CreatePlatformEndpoint* creates an endpoint for a device and mobile app on one of the supported push notification services, such as GCM and APNS. *CreatePlatformEndpoint* requires the PlatformApplicationArn that is returned from *CreatePlatformApplication*. The EndpointArn that is returned when using *CreatePlatformEndpoint* can then be used by the *Publish* action to send a message to a mobile app or by the *Subscribe* action for subscription to a topic. For more information, see [‘Using Amazon SNS Mobile Push Notifications’_](#).

Parameters

- **platform_application_arn** (*string*) – PlatformApplicationArn returned from CreatePlatformApplication is used to create an endpoint.
- **token** (*string*) – Unique identifier created by the notification service for an app on a device. The specific name for Token will vary, depending on which notification service is being used. For example, when using APNS as the notification service, you need the device token. Alternatively, when using GCM or ADM, the device token equivalent is called the registration ID.
- **custom_user_data** (*string*) – Arbitrary user data to associate with the endpoint. SNS does not use this data. The data must be in UTF-8 format and less

than 2KB.

- **attributes** (*map*) – For a list of attributes, see ‘[SetEndpointAttributes](#)’_.

create_topic (*topic*)

Create a new Topic.

Parameters **topic** (*string*) – The name of the new topic.

delete_endpoint (*endpoint_arn=None*)

The *DeleteEndpoint* action, which is idempotent, deletes the endpoint from SNS. For more information, see ‘[Using Amazon SNS Mobile Push Notifications](#)’_.

Parameters **endpoint_arn** (*string*) – EndpointArn of endpoint to delete.

delete_platform_application (*platform_application_arn=None*)

The *DeletePlatformApplication* action deletes a platform application object for one of the supported push notification services, such as APNS and GCM. For more information, see ‘[Using Amazon SNS Mobile Push Notifications](#)’_.

Parameters **platform_application_arn** (*string*) – PlatformApplicationArn of platform application object to delete.

delete_topic (*topic*)

Delete an existing topic

Parameters **topic** (*string*) – The ARN of the topic

get_all_subscriptions (*next_token=None*)

Get list of all subscriptions.

Parameters **next_token** (*string*) – Token returned by the previous call to this method.

get_all_subscriptions_by_topic (*topic, next_token=None*)

Get list of all subscriptions to a specific topic.

Parameters

- **topic** (*string*) – The ARN of the topic for which you wish to find subscriptions.
- **next_token** (*string*) – Token returned by the previous call to this method.

get_all_topics (*next_token=None*)

Parameters **next_token** (*string*) – Token returned by the previous call to this method.

get_endpoint_attributes (*endpoint_arn=None*)

The *GetEndpointAttributes* retrieves the endpoint attributes for a device on one of the supported push notification services, such as GCM and APNS. For more information, see ‘[Using Amazon SNS Mobile Push Notifications](#)’_.

Parameters **endpoint_arn** (*string*) – EndpointArn for GetEndpointAttributes input.

get_platform_application_attributes (*platform_application_arn=None*)

The *GetPlatformApplicationAttributes* action retrieves the attributes of the platform application object for the supported push notification services, such as APNS and GCM. For more information, see ‘[Using Amazon SNS Mobile Push Notifications](#)’_.

Parameters **platform_application_arn** (*string*) – PlatformApplicationArn for GetPlatformApplicationAttributesInput.

get_topic_attributes (*topic*)

Get attributes of a Topic

Parameters **topic** (*string*) – The ARN of the topic.

list_endpoints_by_platform_application (*platform_application_arn=None, next_token=None*)

The *ListEndpointsByPlatformApplication* action lists the endpoints and endpoint attributes for devices

in a supported push notification service, such as GCM and APNS. The results for *ListEndpointsByPlatformApplication* are paginated and return a limited list of endpoints, up to 100. If additional records are available after the first page results, then a `NextToken` string will be returned. To receive the next page, you call *ListEndpointsByPlatformApplication* again using the `NextToken` string received from the previous call. When there are no more records to return, `NextToken` will be null. For more information, see [‘Using Amazon SNS Mobile Push Notifications’](#).

Parameters

- **platform_application_arn** (*string*) – PlatformApplicationArn for ListEndpointsByPlatformApplicationInput action.
- **next_token** (*string*) – NextToken string is used when calling ListEndpointsByPlatformApplication action to retrieve additional records that are available after the first page results.

list_platform_applications (*next_token=None*)

The *ListPlatformApplications* action lists the platform application objects for the supported push notification services, such as APNS and GCM. The results for *ListPlatformApplications* are paginated and return a limited list of applications, up to 100. If additional records are available after the first page results, then a `NextToken` string will be returned. To receive the next page, you call *ListPlatformApplications* using the `NextToken` string received from the previous call. When there are no more records to return, `NextToken` will be null. For more information, see [‘Using Amazon SNS Mobile Push Notifications’](#).

Parameters **next_token** (*string*) – NextToken string is used when calling ListPlatformApplications action to retrieve additional records that are available after the first page results.

publish (*topic=None, message=None, subject=None, target_arn=None, message_structure=None, message_attributes=None*)

Sends a message to all of a topic’s subscribed endpoints

Parameters

- **topic** (*string*) – The topic you want to publish to.
- **message** (*string*) – The message you want to send to the topic. Messages must be UTF-8 encoded strings and be at most 4KB in size.
- **message_structure** (*string*) – Optional parameter. If left as `None`, plain text will be sent. If set to `json`, your message should be a JSON string that matches the structure described at <http://docs.aws.amazon.com/sns/latest/dg/PublishTopic.html#sns-message-formatting-by-protocol>
- **message_attributes** (*dict*) – Message attributes to set. Should be of the form:

```
{
  "name1": {
    "data_type": "Number",
    "string_value": "42"
  },
  "name2": {
    "data_type": "String",
    "string_value": "Bob"
  }
}
```

- **subject** (*string*) – Optional parameter to be used as the “Subject” line of the email notifications.
- **target_arn** (*string*) – Optional parameter for either TopicArn or EndpointArn, but not both.

remove_permission (*topic*, *label*)

Removes a statement from a topic's access control policy.

Parameters

- **topic** (*string*) – The ARN of the topic.
- **label** (*string*) – A unique identifier for the policy statement to be removed.

set_endpoint_attributes (*endpoint_arn=None*, *attributes=None*)

The *SetEndpointAttributes* action sets the attributes for an endpoint for a device on one of the supported push notification services, such as GCM and APNS. For more information, see [‘Using Amazon SNS Mobile Push Notifications’](#).

Parameters

- **endpoint_arn** (*string*) – EndpointArn used for SetEndpointAttributes action.
- **attributes** (*map*) –

A map of the endpoint attributes. Attributes in this map include the following:

- **CustomUserData** – arbitrary user data to associate with the endpoint. SNS does not use this data. The data must be in UTF-8 format and less than 2KB.
- **Enabled** – flag that enables/disables delivery to the endpoint. Message Processor will set this to false when a notification service indicates to SNS that the endpoint is invalid. Users can set it back to true, typically after updating Token.
- **Token** – device token, also referred to as a registration id, for an app and mobile device. This is returned from the notification service when an app and mobile device are registered with the notification service.

set_platform_application_attributes (*platform_application_arn=None*, *attributes=None*)

The *SetPlatformApplicationAttributes* action sets the attributes of the platform application object for the supported push notification services, such as APNS and GCM. For more information, see [‘Using Amazon SNS Mobile Push Notifications’](#).

Parameters

- **platform_application_arn** (*string*) – PlatformApplicationArn for SetPlatformApplicationAttributes action.
- **attributes** (*map*) –

A map of the platform application attributes. Attributes in this map include the following:

- **PlatformCredential** – The credential received from the notification service. For APNS/APNS_SANDBOX, PlatformCredential is “private key”. For GCM, PlatformCredential is “API key”. For ADM, PlatformCredential is “client secret”.
- **PlatformPrincipal** – The principal received from the notification service. For APNS/APNS_SANDBOX, PlatformPrincipal is “SSL certificate”. For GCM, PlatformPrincipal is not applicable. For ADM, PlatformPrincipal is “client id”.
- **EventEndpointCreated** – Topic ARN to which EndpointCreated event notifications should be sent.
- **EventEndpointDeleted** – Topic ARN to which EndpointDeleted event notifications should be sent.
- **EventEndpointUpdated** – Topic ARN to which EndpointUpdate event notifications should be sent.
- **EventDeliveryFailure** – Topic ARN to which DeliveryFailure event notifications should be sent upon Direct Publish delivery failure (permanent) to one of the application's endpoints.

set_topic_attributes (*topic*, *attr_name*, *attr_value*)

Get attributes of a Topic

Parameters

- **topic** (*string*) – The ARN of the topic.
- **attr_name** (*string*) – The name of the attribute you want to set. Only a subset of the topic’s attributes are mutable. Valid values: Policy | DisplayName
- **attr_value** (*string*) – The new value for the attribute.

subscribe (*topic, protocol, endpoint*)

Subscribe to a Topic.

Parameters

- **topic** (*string*) – The ARN of the new topic.
- **protocol** (*string*) – The protocol used to communicate with the subscriber. Current choices are: email|email-json|http|https|sqs|sms|application
- **endpoint** (*string*) – The location of the endpoint for the subscriber. * For email, this would be a valid email address * For email-json, this would be a valid email address * For http, this would be a URL beginning with http * For https, this would be a URL beginning with https * For sqs, this would be the ARN of an SQS Queue * For sms, this would be a phone number of an

SMS-enabled device

– For application, the endpoint is the EndpointArn of a mobile app and device.

subscribe_sqs_queue (*topic, queue*)

Subscribe an SQS queue to a topic.

This is convenience method that handles most of the complexity involved in using an SQS queue as an endpoint for an SNS topic. To achieve this the following operations are performed:

- The correct ARN is constructed for the SQS queue and that ARN is then subscribed to the topic.
- A JSON policy document is constructed that grants permission to the SNS topic to send messages to the SQS queue.
- This JSON policy is then associated with the SQS queue using the queue’s set_attribute method. If the queue already has a policy associated with it, this process will add a Statement to that policy. If no policy exists, a new policy will be created.

Parameters

- **topic** (*string*) – The ARN of the new topic.
- **queue** (*A boto Queue object*) – The queue you wish to subscribe to the SNS Topic.

unsubscribe (*subscription*)

Allows endpoint owner to delete subscription. Confirmation message will be delivered.

Parameters **subscription** (*string*) – The ARN of the subscription to be deleted.

4.138.14 SQS

boto.sqs

`boto.sqs.connect_to_region` (*region_name, **kw_params*)

`boto.sqs.regions` ()

Get all available regions for the SQS service.

Return type *list*

Returns A list of `boto.sqs.regioninfo.RegionInfo`

boto.sqs.attributes

Represents an SQS Attribute Name/Value set

class `boto.sqs.attributes.Attributes` (*parent*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.sqs.connection

class `boto.sqs.connection.SQSConnection` (*aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, proxy_user=None, proxy_pass=None, debug=0, https_connection_factory=None, region=None, path='/', security_token=None, validate_certs=True, profile_name=None*)

A Connection to the SQS Service.

APIVersion = '2012-11-05'

AuthServiceName = 'sqs'

DefaultContentType = 'text/plain'

DefaultRegionEndpoint = 'queue.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.SQSError`

add_permission (*queue, label, aws_account_id, action_name*)

Add a permission to a queue.

Parameters

- **queue** (*boto.sqs.queue.Queue*) – The queue object
- **label** (*str or unicode*) – A unique identification of the permission you are setting. Maximum of 80 characters [0-9a-zA-Z_-] Example, AliceSendMessage
- **principal_id** – The AWS account number of the principal who will be given permission. The principal must have an AWS account, but does not need to be signed up for Amazon SQS. For information about locating the AWS account identification.
- **action_name** (*str or unicode*) – The action. Valid choices are: * * * SendMessage * ReceiveMessage * DeleteMessage * ChangeMessageVisibility * GetQueueAttributes

Return type `bool`

Returns True if successful, False otherwise.

change_message_visibility (*queue, receipt_handle, visibility_timeout*)

Extends the read lock timeout for the specified message from the specified queue to the specified value.

Parameters

- **queue** (A *boto.sqs.queue.Queue* object) – The Queue from which messages are read.
- **receipt_handle** (*str*) – The receipt handle associated with the message whose visibility timeout will be changed.
- **visibility_timeout** (*int*) – The new value of the message’s visibility timeout in seconds.

change_message_visibility_batch (*queue, messages*)

A batch version of `change_message_visibility` that can act on up to 10 messages at a time.

Parameters

- **queue** (A *boto.sqs.queue.Queue* object.) – The Queue to which the messages will be written.
- **messages** (*List of tuples.*) – A list of tuples where each tuple consists of a *boto.sqs.message.Message* object and an integer that represents the new visibility timeout for that message.

create_queue (*queue_name, visibility_timeout=None*)

Create an SQS Queue.

Parameters

- **queue_name** (*str or unicode*) – The name of the new queue. Names are scoped to an account and need to be unique within that account. Calling this method on an existing queue name will not return an error from SQS unless the value for `visibility_timeout` is different than the value of the existing queue of that name. This is still an expensive operation, though, and not the preferred way to check for the existence of a queue. See the *boto.sqs.connection.SQSConnection.lookup()* method.
- **visibility_timeout** (*int*) – The default visibility timeout for all messages written in the queue. This can be overridden on a per-message.

Return type *boto.sqs.queue.Queue*

Returns The newly created queue.

delete_message (*queue, message*)

Delete a message from a queue.

Parameters

- **queue** (A *boto.sqs.queue.Queue* object) – The Queue from which messages are read.
- **message** (A *boto.sqs.message.Message* object) – The Message to be deleted

Return type *bool*

Returns True if successful, False otherwise.

delete_message_batch (*queue, messages*)

Deletes a list of messages from a queue in a single request.

Parameters

- **queue** (A *boto.sqs.queue.Queue* object.) – The Queue to which the messages will be written.

- **messages** (List of *boto.sqs.message.Message* objects.) – A list of message objects.

delete_message_from_handle (*queue, receipt_handle*)

Delete a message from a queue, given a receipt handle.

Parameters

- **queue** (A *boto.sqs.queue.Queue* object) – The Queue from which messages are read.
- **receipt_handle** (*str*) – The receipt handle for the message

Return type *bool*

Returns True if successful, False otherwise.

delete_queue (*queue, force_deletion=False*)

Delete an SQS Queue.

Parameters

- **queue** (A *Queue* object) – The SQS queue to be deleted
- **force_deletion** (*Boolean*) – A deprecated parameter that is no longer used by SQS's API.

Return type *bool*

Returns True if the command succeeded, False otherwise

get_all_queues (*prefix=""*)

Retrieves all queues.

Parameters **prefix** (*str*) – Optionally, only return queues that start with this value.

Return type *list*

Returns A list of *boto.sqs.queue.Queue* instances.

get_dead_letter_source_queues (*queue*)

Retrieves the dead letter source queues for a given queue.

Parameters **queue** (A *boto.sqs.queue.Queue* object.) – The queue for which to get DL source queues

Return type *list*

Returns A list of *boto.sqs.queue.Queue* instances.

get_queue (*queue_name, owner_acct_id=None*)

Retrieves the queue with the given name, or None if no match was found.

Parameters

- **queue_name** (*str*) – The name of the queue to retrieve.
- **owner_acct_id** (*str*) – Optionally, the AWS account ID of the account that created the queue.

Return type *boto.sqs.queue.Queue* or None

Returns The requested queue, or None if no match was found.

get_queue_attributes (*queue, attribute='All'*)

Gets one or all attributes of a Queue

Parameters

- **queue** (A *Queue* object) – The SQS queue to get attributes for

- **attribute** (*str*) – The specific attribute requested. If not supplied, the default is to return all attributes. Valid attributes are:
 - All
 - ApproximateNumberOfMessages
 - ApproximateNumberOfMessagesNotVisible
 - VisibilityTimeout
 - CreatedTimestamp
 - LastModifiedTimestamp
 - Policy
 - MaximumMessageSize
 - MessageRetentionPeriod
 - QueueArn
 - ApproximateNumberOfMessagesDelayed
 - DelaySeconds
 - ReceiveMessageWaitTimeSeconds
 - RedrivePolicy

Return type *boto.sqs.attributes.Attributes*

Returns An Attributes object containing request value(s).

lookup (*queue_name*, *owner_acct_id=None*)

Retrieves the queue with the given name, or None if no match was found.

Parameters

- **queue_name** (*str*) – The name of the queue to retrieve.
- **owner_acct_id** (*str*) – Optionally, the AWS account ID of the account that created the queue.

Return type *boto.sqs.queue.Queue* or None

Returns The requested queue, or None if no match was found.

purge_queue (*queue*)

Purge all messages in an SQS Queue.

Parameters **queue** (*A Queue object*) – The SQS queue to be purged

Return type *bool*

Returns True if the command succeeded, False otherwise

receive_message (*queue*, *number_messages=1*, *visibility_timeout=None*, *attributes=None*,
wait_time_seconds=None, *message_attributes=None*)

Read messages from an SQS Queue.

Parameters

- **queue** (*A Queue object*) – The Queue from which messages are read.
- **number_messages** (*int*) – The maximum number of messages to read (default=1)

- **visibility_timeout** (*int*) – The number of seconds the message should remain invisible to other queue readers (default=None which uses the Queues default)
- **attributes** (*str*) – The name of additional attribute to return with response or All if you want all attributes. The default is to return no additional attributes. Valid values: * All * SenderId * SentTimestamp * ApproximateReceiveCount * ApproximateFirstReceiveTimestamp
- **wait_time_seconds** (*int*) – The duration (in seconds) for which the call will wait for a message to arrive in the queue before returning. If a message is available, the call will return sooner than wait_time_seconds.
- **message_attributes** (*list*) – The name(s) of additional message attributes to return. The default is to return no additional message attributes. Use ['All'] or ['. * '] to return all.

Return type `list`

Returns A list of `boto.sqs.message.Message` objects.

remove_permission (*queue, label*)

Remove a permission from a queue.

Parameters

- **queue** (`boto.sqs.queue.Queue`) – The queue object
- **label** (*str* or *unicode*) – The unique label associated with the permission being removed.

Return type `bool`

Returns True if successful, False otherwise.

send_message (*queue, message_content, delay_seconds=None, message_attributes=None*)

Send a new message to the queue.

Parameters

- **queue** (A `boto.sqs.queue.Queue` object.) – The Queue to which the messages will be written.
- **message_content** (*string*) – The body of the message
- **delay_seconds** (*int*) – Number of seconds (0 - 900) to delay this message from being processed.
- **message_attributes** (*dict*) – Message attributes to set. Should be of the form:

```
{
    "name1": { "data_type": "Number", "string_value": "1"
    }, "name2": {
        "data_type": "String", "string_value": "Bob"
    }
}
```

send_message_batch (*queue, messages*)

Delivers up to 10 messages to a queue in a single request.

Parameters

- **queue** (A `boto.sqs.queue.Queue` object.) – The Queue to which the messages will be written.
- **messages** (*List of lists.*) – A list of lists or tuples. Each inner tuple represents a single message to be written and consists of an ID (string) that must be unique within the list of messages, the message body itself which can be a maximum of 64K in length, an integer which represents the delay time (in seconds) for the message (0-900) before the message will be delivered to the queue, and an optional dict of message attributes like those passed to `send_message` above.

set_queue_attribute (*queue, attribute, value*)

Set a new value for an attribute of a Queue.

Parameters

- **queue** (A *Queue object*) – The SQS queue to get attributes for
- **attribute** (*String*) – The name of the attribute you want to set.
- **value** – The new value for the attribute must be:
 - For *DelaySeconds* the value must be an integer number of

seconds from 0 to 900 (15 minutes).

```
>>> connection.set_queue_attribute(queue, 'DelaySeconds
↳', 900)
```

- For *MaximumMessageSize* the value must be an integer number of

bytes from 1024 (1 KiB) to 262144 (256 KiB).

```
>>> connection.set_queue_attribute(queue,
↳ 'MaximumMessageSize', 262144)
```

- For *MessageRetentionPeriod* the value must be an integer number of

seconds from 60 (1 minute) to 1209600 (14 days).

```
>>> connection.set_queue_attribute(queue,
↳ 'MessageRetentionPeriod', 1209600)
```

- For *Policy* the value must be a string that contains JSON formatted

parameters and values.

```
>>> connection.set_queue_attribute(queue, 'Policy',
↳ json.dumps({
...     'Version': '2008-10-17',
...     'Id': '/123456789012/testQueue/SQSDefaultPolicy
↳',
...     'Statement': [
...         {
...             'Sid': 'Queue1ReceiveMessage',
...             'Effect': 'Allow',
...             'Principal': {
...                 'AWS': '*'
```

(continues on next page)

(continued from previous page)

```

...         },
...         'Action': 'SQS:ReceiveMessage',
...         'Resource': 'arn:aws:aws:sqs:us-east-
↪1:123456789012:testQueue'
...     }
... ]
... )))

```

- For *ReceiveMessageWaitTimeSeconds* the value must be an integer number of

seconds from 0 to 20.

```

>>> connection.set_queue_attribute(queue,
↪'ReceiveMessageWaitTimeSeconds', 20)

```

- For *VisibilityTimeout* the value must be an integer number of

seconds from 0 to 43200 (12 hours).

```

>>> connection.set_queue_attribute(queue,
↪'VisibilityTimeout', 43200)

```

- For *RedrivePolicy* the value must be a string that contains JSON formatted parameters and values. You can set *maxReceiveCount* to a value between 1 and 1000. The *deadLetterTargetArn* value is the Amazon Resource Name (ARN) of the queue that will receive the dead letter messages.

```

>>> connection.set_queue_attribute(queue, 'RedrivePolicy',
↪json.dumps({
...     'maxReceiveCount': 5,
...     'deadLetterTargetArn': "arn:aws:aws:sqs:us-east-
↪1:123456789012:testDeadLetterQueue"
... })))

```

boto.sqs.jsonmessage

class boto.sqs.jsonmessage.**JSONMessage** (*queue=None, body=None, xml_attrs=None*)
Acts like a dictionary but encodes it's data as a Base64 encoded JSON payload.

decode (*value*)

Transform seralized byte array into any object.

encode (*value*)

Transform body object into serialized byte array format.

boto.sqs.message

SQS Message

A Message represents the data stored in an SQS queue. The rules for what is allowed within an SQS Message are here:

http://docs.amazonwebservices.com/AWSSimpleQueueService/2008-01-01/SQSDeveloperGuide/Query_QuerySendMessage.html

So, at it's simplest level a Message just needs to allow a developer to store bytes in it and get the bytes back out. However, to allow messages to have richer semantics, the Message class must support the following interfaces:

The constructor for the Message class must accept a keyword parameter "queue" which is an instance of a boto Queue object and represents the queue that the message will be stored in. The default value for this parameter is None.

The constructor for the Message class must accept a keyword parameter "body" which represents the content or body of the message. The format of this parameter will depend on the behavior of the particular Message subclass. For example, if the Message subclass provides dictionary-like behavior to the user the body passed to the constructor should be a dict-like object that can be used to populate the initial state of the message.

The Message class must provide an encode method that accepts a value of the same type as the body parameter of the constructor and returns a string of characters that are able to be stored in an SQS message body (see rules above).

The Message class must provide a decode method that accepts a string of characters that can be stored (and probably were stored!) in an SQS message and return an object of a type that is consistent with the "body" parameter accepted on the class constructor.

The Message class must provide a `__len__` method that will return the size of the encoded message that would be stored in SQS based on the current state of the Message object.

The Message class must provide a `get_body` method that will return the body of the message in the same format accepted in the constructor of the class.

The Message class must provide a `set_body` method that accepts a message body in the same format accepted by the constructor of the class. This method should alter to the internal state of the Message object to reflect the state represented in the message body parameter.

The Message class must provide a `get_body_encoded` method that returns the current body of the message in the format in which it would be stored in SQS.

class boto.sqs.message.**EncodedMHMessage** (*queue=None, body=None, xml_attrs=None*)

The EncodedMHMessage class provides a message that provides RFC821-like headers like this:

HeaderName: HeaderValue

This variation encodes/decodes the body of the message in base64 automatically. The message instance can be treated like a mapping object, i.e. `m['HeaderName']` would return 'HeaderValue'.

decode (*value*)

Transform serialized byte array into any object.

encode (*value*)

Transform body object into serialized byte array format.

class boto.sqs.message.**MHMessage** (*queue=None, body=None, xml_attrs=None*)

The MHMessage class provides a message that provides RFC821-like headers like this:

HeaderName: HeaderValue

The encoding/decoding of this is handled automatically and after the message body has been read, the message instance can be treated like a mapping object, i.e. `m['HeaderName']` would return 'HeaderValue'.

decode (*value*)

Transform serialized byte array into any object.

encode (*value*)

Transform body object into serialized byte array format.

get (*key, default=None*)

has_key (*key*)**items** ()**keys** ()**update** (*d*)**values** ()**class** boto.sqs.message.**Message** (*queue=None, body=""*)

The default Message class used for SQS queues. This class automatically encodes/decodes the message body using Base64 encoding to avoid any illegal characters in the message body. See:

<https://forums.aws.amazon.com/thread.jspa?threadID=13067>

for details on why this is a good idea. The encode/decode is meant to be transparent to the end-user.

decode (*value*)

Transform serialized byte array into any object.

encode (*value*)

Transform body object into serialized byte array format.

class boto.sqs.message.**RawMessage** (*queue=None, body=""*)

Base class for SQS messages. RawMessage does not encode the message in any way. Whatever you store in the body of the message is what will be written to SQS and whatever is returned from SQS is stored directly into the body of the message.

change_visibility (*visibility_timeout*)**decode** (*value*)

Transform serialized byte array into any object.

delete ()**encode** (*value*)

Transform body object into serialized byte array format.

endElement (*name, value, connection*)**endNode** (*connection*)**get_body** ()**get_body_encoded** ()

This method is really a semi-private method used by the Queue.write method when writing the contents of the message to SQS. You probably shouldn't need to call this method in the normal course of events.

set_body (*body*)

Override the current body for this object, using decoded format.

startElement (*name, attrs, connection*)

boto.sqs.queue

Represents an SQS Queue

class boto.sqs.queue.**Queue** (*connection=None, url=None, message_class=<class 'boto.sqs.message.Message'>*)

add_permission (*label, aws_account_id, action_name*)

Add a permission to a queue.

Parameters

- **label** (*str or unicode*) – A unique identification of the permission you are setting. Maximum of 80 characters [0-9a-zA-Z_-] Example, Alice-SendMessage
- **principal_id** – The AWS account number of the principal who will be given permission. The principal must have an AWS account, but does not need to be signed up for Amazon SQS. For information about locating the AWS account identification.
- **action_name** (*str or unicode*) – The action. Valid choices are: SendMessage|ReceiveMessage|DeleteMessage| ChangeMessageVisibility|GetQueueAttributes*

Return type `bool`

Returns True if successful, False otherwise.

arn

change_message_visibility_batch (*messages*)

A batch version of `change_message_visibility` that can act on up to 10 messages at a time.

Parameters *messages* (*List of tuples.*) – A list of tuples where each tuple consists of a `boto.sqs.message.Message` object and an integer that represents the new visibility timeout for that message.

clear (*page_size=10, vtimeout=10*)

Deprecated utility function to remove all messages from a queue

count (*page_size=10, vtimeout=10*)

Utility function to count the number of messages in a queue. Note: This function now calls `GetQueueAttributes` to obtain an ‘approximate’ count of the number of messages in a queue.

count_slow (*page_size=10, vtimeout=10*)

Deprecated. This is the old ‘count’ method that actually counts the messages by reading them all. This gives an accurate count but is very slow for queues with non-trivial number of messages. Instead, use `get_attributes(‘ApproximateNumberOfMessages’)` to take advantage of the new SQS capability. This is retained only for the unit tests.

delete ()

Delete the queue.

delete_message (*message*)

Delete a message from the queue.

Parameters *message* (`boto.sqs.message.Message`) – The `boto.sqs.message.Message` object to delete.

Return type `bool`

Returns True if successful, False otherwise

delete_message_batch (*messages*)

Deletes a list of messages in a single request.

Parameters *messages* (*List of boto.sqs.message.Message objects.*) – A list of message objects.

dump (*file_name, page_size=10, vtimeout=10, sep=‘\n’*)

Utility function to dump the messages in a queue to a file NOTE: Page size must be < 10 else SQS errors

endElement (*name, value, connection*)

get_attributes (*attributes='All'*)

Retrieves attributes about this queue object and returns them in an Attribute instance (subclass of a Dictionary).

Parameters **attributes** (*string*) – String containing one of: ApproximateNumberOfMessages, ApproximateNumberOfMessagesNotVisible, VisibilityTimeout, CreatedTimestamp, LastModifiedTimestamp, Policy ReceiveMessageWaitTimeSeconds

Return type Attribute object

Returns An Attribute object which is a mapping type holding the requested name/value pairs

get_messages (*num_messages=1, visibility_timeout=None, attributes=None, wait_time_seconds=None, message_attributes=None*)

Get a variable number of messages.

Parameters

- **num_messages** (*int*) – The maximum number of messages to read from the queue.
- **visibility_timeout** (*int*) – The VisibilityTimeout for the messages read.
- **attributes** (*str*) – The name of additional attribute to return with response or All if you want all attributes. The default is to return no additional attributes. Valid values: All SenderId SentTimestamp ApproximateReceiveCount ApproximateFirstReceiveTimestamp
- **wait_time_seconds** (*int*) – The duration (in seconds) for which the call will wait for a message to arrive in the queue before returning. If a message is available, the call will return sooner than wait_time_seconds.
- **message_attributes** (*list*) – The name(s) of additional message attributes to return. The default is to return no additional message attributes. Use ['All'] or ['. * '] to return all.

Return type list

Returns A list of *boto.sqs.message.Message* objects.

get_timeout ()

Get the visibility timeout for the queue.

Return type int

Returns The number of seconds as an integer.

id

load (*file_name, sep='\n'*)

Utility function to load messages from a local filename to a queue

load_from_file (*fp, sep='\n'*)

Utility function to load messages from a file-like object to a queue

load_from_filename (*file_name, sep='\n'*)

Utility function to load messages from a local filename to a queue

load_from_s3 (*bucket, prefix=None*)

Load messages previously saved to S3.

name

new_message (*body="", **kwargs*)

Create new message of appropriate class.

Parameters **body** (*message body*) – The body of the newly created message (optional).

Return type *boto.sqs.message.Message*

Returns A new Message object

purge ()

Purge all messages in the queue.

read (*visibility_timeout=None, wait_time_seconds=None, message_attributes=None*)

Read a single message from the queue.

Parameters

- **visibility_timeout** (*int*) – The timeout for this message in seconds
- **wait_time_seconds** (*int*) – The duration (in seconds) for which the call will wait for a message to arrive in the queue before returning. If a message is available, the call will return sooner than `wait_time_seconds`.
- **message_attributes** (*list*) – The name(s) of additional message attributes to return. The default is to return no additional message attributes. Use `['All']` or `['.*']` to return all.

Return type *boto.sqs.message.Message*

Returns A single message or None if queue is empty

remove_permission (*label*)

Remove a permission from a queue.

Parameters **label** (*str or unicode*) – The unique label associated with the permission being removed.

Return type *bool*

Returns True if successful, False otherwise.

save (*file_name, sep='\n'*)

Read all messages from the queue and persist them to local file. Messages are written to the file and the 'sep' string is written in between messages. Messages are deleted from the queue after being written to the file. Returns the number of messages saved.

save_to_file (*fp, sep='\n'*)

Read all messages from the queue and persist them to file-like object. Messages are written to the file and the 'sep' string is written in between messages. Messages are deleted from the queue after being written to the file. Returns the number of messages saved.

save_to_filename (*file_name, sep='\n'*)

Read all messages from the queue and persist them to local file. Messages are written to the file and the 'sep' string is written in between messages. Messages are deleted from the queue after being written to the file. Returns the number of messages saved.

save_to_s3 (*bucket*)

Read all messages from the queue and persist them to S3. Messages are stored in the S3 bucket using a naming scheme of:

`<queue_id>/<message_id>`

Messages are deleted from the queue after being saved to S3. Returns the number of messages saved.

set_attribute (*attribute, value*)

Set a new value for an attribute of the Queue.

Parameters

- **attribute** (*String*) – The name of the attribute you want to set.

- **value** – The new value for the attribute must be:
 - For *DelaySeconds* the value must be an integer number of

seconds from 0 to 900 (15 minutes).

```
>>> queue.set_attribute('DelaySeconds', 900)
```

- For *MaximumMessageSize* the value must be an integer number of

bytes from 1024 (1 KiB) to 262144 (256 KiB).

```
>>> queue.set_attribute('MaximumMessageSize', 262144)
```

- For *MessageRetentionPeriod* the value must be an integer number of

seconds from 60 (1 minute) to 1209600 (14 days).

```
>>> queue.set_attribute('MessageRetentionPeriod', 1209600)
```

- For *Policy* the value must be a string that contains JSON formatted

parameters and values.

```
>>> queue.set_attribute('Policy', json.dumps({
...     'Version': '2008-10-17',
...     'Id': '/123456789012/testQueue/SQSDefaultPolicy',
...     'Statement': [
...         {
...             'Sid': 'Queue1ReceiveMessage',
...             'Effect': 'Allow',
...             'Principal': {
...                 'AWS': '*'
...             },
...             'Action': 'SQS:ReceiveMessage',
...             'Resource': 'arn:aws:aws:sqs:us-east-1:123456789012:testQueue'
...         }
...     ]
... }))
```

- For *ReceiveMessageWaitTimeSeconds* the value must be an integer number of

seconds from 0 to 20.

```
>>> queue.set_attribute('ReceiveMessageWaitTimeSeconds', 20)
```

- For *VisibilityTimeout* the value must be an integer number of

seconds from 0 to 43200 (12 hours).

```
>>> queue.set_attribute('VisibilityTimeout', 43200)
```

– For *RedrivePolicy* the value must be an string that contains JSON formatted

parameters and values. You can set `maxReceiveCount` to a value between 1 and 1000. The `deadLetterTargetArn` value is the Amazon Resource Name (ARN) of the queue that will receive the dead letter messages.

```
>>> queue.set_attribute('RedrivePolicy', json.dumps({
...     'maxReceiveCount': 5,
...     'deadLetterTargetArn': "arn:aws:sqs:us-east-
→1:123456789012:testDeadLetterQueue"
... }))
```

Return type `bool`

Returns True if successful, otherwise False.

set_message_class (*message_class*)

Set the message class that should be used when instantiating messages read from the queue. By default, the class `boto.sqs.message.Message` is used but this can be overridden with any class that behaves like a message.

Parameters `message_class` (*Message-like class*) – The new Message class

set_timeout (*visibility_timeout*)

Set the visibility timeout for the queue.

Parameters `visibility_timeout` (*int*) – The desired timeout in seconds

startElement (*name, attrs, connection*)

write (*message, delay_seconds=None*)

Add a single message to the queue.

Parameters `message` (*Message*) – The message to be written to the queue

Return type `boto.sqs.message.Message`

Returns The `boto.sqs.message.Message` object that was written.

write_batch (*messages*)

Delivers up to 10 messages in a single request.

Parameters `messages` (*List of lists.*) – A list of lists or tuples. Each inner tuple represents a single message to be written and consists of and ID (string) that must be unique within the list of messages, the message body itself which can be a maximum of 64K in length, an integer which represents the delay time (in seconds) for the message (0-900) before the message will be delivered to the queue, and an optional dict of message attributes like those passed to `send_message` in the connection class.

boto.sqs.regioninfo

```
class boto.sqs.regioninfo.SQSRegionInfo (connection=None, name=None, endpoint=None,
                                         connection_cls=None)
```

boto.sqs.batchresults

A set of results returned by `SendMessageBatch`.

class boto.sqs.batchresults.**BatchResults** (*parent*)

A container for the results of a `send_message_batch` request.

Variables

- **results** – A list of successful results. Each item in the list will be an instance of `ResultEntry`.
- **errors** – A list of unsuccessful results. Each item in the list will be an instance of `ResultEntry`.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.sqs.batchresults.**ResultEntry**

The result (successful or unsuccessful) of a single message within a `send_message_batch` request.

In the case of a successful result, this dict-like object will contain the following items:

Variables

- **id** – A string containing the user-supplied ID of the message.
- **message_id** – A string containing the SQS ID of the new message.
- **message_md5** – A string containing the MD5 hash of the message body.

In the case of an error, this object will contain the following items:

Variables

- **id** – A string containing the user-supplied ID of the message.
- **sender_fault** – A boolean value.
- **error_code** – A string containing a short description of the error.
- **error_message** – A string containing a description of the error.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.138.15 STS

boto.sts

`boto.sts.connect_to_region` (*region_name, **kw_params*)

Given a valid region name, return a `boto.sts.connection.STSConnection`.

Type str

Parameters **region_name** – The name of the region to connect to.

Return type `boto.sts.connection.STSConnection` or None

Returns A connection to the given region, or None if an invalid region name is given

`boto.sts.regions` ()

Get all available regions for the STS service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo` instances

class boto.sts.**STSConnection** (*aws_access_key_id=None, aws_secret_access_key=None, is_secure=True, port=None, proxy=None, proxy_port=None, proxy_user=None, proxy_pass=None, debug=0, https_connection_factory=None, region=None, path='/', converter=None, validate_certs=True, anon=False, security_token=None, profile_name=None*)

AWS Security Token Service The AWS Security Token Service is a web service that enables you to request

temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users). This guide provides descriptions of the AWS Security Token Service API.

For more detailed information about using this service, go to [‘Using Temporary Security Credentials’](#).

For information about setting up signatures and authorization through the API, go to [‘Signing AWS API Requests’](#) in the AWS General Reference . For general information about the Query API, go to [‘Making Query Requests’](#) in Using IAM . For information about using security tokens with other AWS products, go to [‘Using Temporary Security Credentials to Access AWS’](#) in Using Temporary Security Credentials .

If you’re new to AWS and need additional technical information about a specific AWS product, you can find the product’s technical documentation at [‘http://aws.amazon.com/documentation/’](http://aws.amazon.com/documentation/).

We will refer to Amazon Identity and Access Management using the abbreviated form IAM. All copyrights and legal protections still apply.

Parameters `anon` (*boolean*) – If this parameter is True, the `STSTokenConnection` object will make anonymous requests, and it will not use AWS Credentials or even search for AWS Credentials to make these requests.

APIVersion = '2011-06-15'

DefaultRegionEndpoint = 'sts.amazonaws.com'

DefaultRegionName = 'us-east-1'

assume_role (*role_arn, role_session_name, policy=None, duration_seconds=None, external_id=None, mfa_serial_number=None, mfa_token=None*)

Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) that you can use to access AWS resources that you might not normally have access to. Typically, you use `AssumeRole` for cross-account access or federation.

For cross-account access, imagine that you own multiple accounts and need to access resources in each account. You could create long-term credentials in each account to access those resources. However, managing all those credentials and remembering which one can access which account can be time consuming. Instead, you can create one set of long-term credentials in one account and then use temporary security credentials to access all the other accounts by assuming roles in those accounts. For more information about roles, see [‘Roles’](#) in Using IAM .

For federation, you can, for example, grant single sign-on access to the AWS Management Console. If you already have an identity and authentication system in your corporate network, you don’t have to recreate user identities in AWS in order to grant those user identities access to AWS. Instead, after a user has been authenticated, you call `AssumeRole` (and specify the role with the appropriate permissions) to get temporary security credentials for that user. With those temporary security credentials, you construct a sign-in URL that users can use to access the console. For more information, see [‘Scenarios for Granting Temporary Access’](#) in AWS Security Token Service .

The temporary security credentials are valid for the duration that you specified when calling `AssumeRole`, which can be from 900 seconds (15 minutes) to 3600 seconds (1 hour). The default is 1 hour.

The temporary security credentials that are returned from the `AssumeRoleWithWebIdentity` response have the permissions that are associated with the access policy of the role being assumed and any policies that are associated with the AWS resource being accessed. You can further restrict the permissions of the temporary security credentials by passing a policy in the request. The resulting permissions are an intersection of the role’s access policy and the policy that you passed. These policies and any applicable resource-based policies are evaluated when calls to AWS service APIs are made using the temporary security credentials.

To assume a role, your AWS account must be trusted by the role. The trust relationship is defined in the role’s trust policy when the IAM role is created. You must also have a policy that allows you to call `sts:AssumeRole`.

Important: You cannot call *Assumerole* by using AWS account credentials; access will be denied. You must use IAM user credentials to call *AssumeRole*.

Parameters

- **role_arn** (*string*) – The Amazon Resource Name (ARN) of the role that the caller is assuming.
- **role_session_name** (*string*) – An identifier for the assumed role session. The session name is included as part of the *AssumedRoleUser*.
- **policy** (*string*) – A supplemental policy that is associated with the temporary security credentials from the *AssumeRole* call. The resulting permissions of the temporary security credentials are an intersection of this policy and the access policy that is associated with the role. Use this policy to further restrict the permissions of the temporary security credentials.
- **duration_seconds** (*integer*) – The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds.
- **external_id** (*string*) – A unique identifier that is used by third parties to assume a role in their customers' accounts. For each role that the third party can assume, they should instruct their customers to create a role with the external ID that the third party generated. Each time the third party assumes the role, they must pass the customer's external ID. The external ID is useful in order to help third parties bind a role to the customer who created it. For more information about the external ID, see [‘About the External ID’](#) in Using Temporary Security Credentials .
- **mfa_serial_number** (*string*) – The identification number of the MFA device that is associated with the user who is making the *AssumeRole* call. Specify this value if the trust policy of the role being assumed includes a condition that requires MFA authentication. The value is either the serial number for a hardware device (such as GAHT12345678) or an Amazon Resource Name (ARN) for a virtual device (such as arn:aws:iam::123456789012:mfa/user). Minimum length of 9. Maximum length of 256.
- **mfa_token** (*string*) – The value provided by the MFA device, if the trust policy of the role being assumed requires MFA (that is, if the policy includes a condition that tests for MFA). If the role being assumed requires MFA and if the *TokenCode* value is missing or expired, the *AssumeRole* call returns an “access denied” error. Minimum length of 6. Maximum length of 6.

assume_role_with_saml (*role_arn*, *principal_arn*, *saml_assertion*, *policy=None*, *duration_seconds=None*)

Returns a set of temporary security credentials for users who have been authenticated via a SAML authentication response. This operation provides a mechanism for tying an enterprise identity store or directory to role-based AWS access without user-specific credentials or configuration.

The temporary security credentials returned by this operation consist of an access key ID, a secret access key, and a security token. Applications can use these temporary security credentials to sign calls to AWS services. The credentials are valid for the duration that you specified when calling *AssumeRoleWithSAML*, which can be up to 3600 seconds (1 hour) or until the time specified in the SAML authentication response's *NotOnOrAfter* value, whichever is shorter.

The maximum duration for a session is 1 hour, and the minimum duration is 15 minutes, even if values outside this range are specified.

Optionally, you can pass an AWS IAM access policy to this operation. The temporary security credentials that are returned by the operation have the permissions that are associated with the access policy of the role

being assumed, except for any permissions explicitly denied by the policy you pass. This gives you a way to further restrict the permissions for the federated user. These policies and any applicable resource-based policies are evaluated when calls to AWS are made using the temporary security credentials.

Before your application can call `AssumeRoleWithSAML`, you must configure your SAML identity provider (IdP) to issue the claims required by AWS. Additionally, you must use AWS Identity and Access Management (AWS IAM) to create a SAML provider entity in your AWS account that represents your identity provider, and create an AWS IAM role that specifies this SAML provider in its trust policy.

Calling `AssumeRoleWithSAML` does not require the use of AWS security credentials. The identity of the caller is validated by using keys in the metadata document that is uploaded for the SAML provider entity for your identity provider.

For more information, see the following resources:

- **‘Creating Temporary Security Credentials for SAML Federation’** in the Using Temporary Security Credentials guide.
- **‘SAML Providers’** in the Using IAM guide.
- **‘Configuring a Relying Party and Claims in the Using IAM guide.’**
- **‘Creating a Role for SAML-Based Federation’** in the Using IAM guide.

Parameters

- **`role_arn`** (*string*) – The Amazon Resource Name (ARN) of the role that the caller is assuming.
- **`principal_arn`** (*string*) – The Amazon Resource Name (ARN) of the SAML provider in AWS IAM that describes the IdP.
- **`saml_assertion`** (*string*) – The base-64 encoded SAML authentication response provided by the IdP.

For more information, see **‘Configuring a Relying Party and Adding Claims’** in the Using IAM guide.

Parameters **`policy`** (*string*) –

An AWS IAM policy in JSON format.

The temporary security credentials that are returned by this operation have the permissions that are associated with the access policy of the role being assumed, except for any permissions explicitly denied by the policy you pass. These policies and any applicable resource-based policies are evaluated when calls to AWS are made using the temporary security credentials.

The policy must be 2048 bytes or shorter, and its packed size must be less than 450 bytes.

Parameters **`duration_seconds`** (*integer*) –

The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds. An expiration can also be specified in the SAML authentication response’s *NotOnOrAfter* value. The actual expiration time is whichever value is shorter.

The maximum duration for a session is 1 hour, and the minimum duration is 15 minutes, even if values outside this range are specified.

```
assume_role_with_web_identity (role_arn, role_session_name, web_identity_token,  
                                provider_id=None, policy=None, duration_seconds=None)
```

Returns a set of temporary security credentials for users who have been authenticated in a mobile or web application with a web identity provider, such as Login with Amazon, Facebook, or Google. `AssumeRoleWithWebIdentity` is an API call that does not require the use of AWS security credentials. Therefore, you can distribute an application (for example, on mobile devices) that requests temporary security credentials without including long-term AWS credentials in the application or by deploying server-based proxy

services that use long-term AWS credentials. For more information, see [‘Creating a Mobile Application with Third-Party Sign-In’](#) in AWS Security Token Service .

The temporary security credentials consist of an access key ID, a secret access key, and a security token. Applications can use these temporary security credentials to sign calls to AWS service APIs. The credentials are valid for the duration that you specified when calling *AssumeRoleWithWebIdentity*, which can be from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the temporary security credentials are valid for 1 hour.

The temporary security credentials that are returned from the *AssumeRoleWithWebIdentity* response have the permissions that are associated with the access policy of the role being assumed. You can further restrict the permissions of the temporary security credentials by passing a policy in the request. The resulting permissions are an intersection of the role’s access policy and the policy that you passed. These policies and any applicable resource-based policies are evaluated when calls to AWS service APIs are made using the temporary security credentials.

Before your application can call *AssumeRoleWithWebIdentity*, you must have an identity token from a supported identity provider and create a role that the application can assume. The role that your application assumes must trust the identity provider that is associated with the identity token. In other words, the identity provider must be specified in the role’s trust policy. For more information, see [‘Creating Temporary Security Credentials for Mobile Apps Using Third-Party Identity Providers’](#).

Parameters

- **role_arn** (*string*) – The Amazon Resource Name (ARN) of the role that the caller is assuming.
- **role_session_name** (*string*) – An identifier for the assumed role session. Typically, you pass the name or identifier that is associated with the user who is using your application. That way, the temporary security credentials that your application will use are associated with that user. This session name is included as part of the ARN and assumed role ID in the *AssumedRoleUser* response element.
- **web_identity_token** (*string*) – The OAuth 2.0 access token or OpenID Connect ID token that is provided by the identity provider. Your application must get this token by authenticating the user who is using your application with a web identity provider before the application makes an *AssumeRoleWithWebIdentity* call.
- **provider_id** (*string*) – Specify this value only for OAuth access tokens. Do not specify this value for OpenID Connect ID tokens, such as *accounts.google.com*. This is the fully-qualified host component of the domain name of the identity provider. Do not include URL schemes and port numbers. Currently, *www.amazon.com* and *graph.facebook.com* are supported.
- **policy** (*string*) – A supplemental policy that is associated with the temporary security credentials from the *AssumeRoleWithWebIdentity* call. The resulting permissions of the temporary security credentials are an intersection of this policy and the access policy that is associated with the role. Use this policy to further restrict the permissions of the temporary security credentials.
- **duration_seconds** (*integer*) – The duration, in seconds, of the role session. The value can range from 900 seconds (15 minutes) to 3600 seconds (1 hour). By default, the value is set to 3600 seconds.

decode_authorization_message (*encoded_message*)

Decodes additional information about the authorization status of a request from an encoded message returned in response to an AWS request.

For example, if a user is not authorized to perform an action that he or she has requested, the request

returns a *Client.UnauthorizedOperation* response (an HTTP 403 response). Some AWS actions additionally return an encoded message that can provide details about this authorization failure. Only certain AWS actions return an encoded authorization message. The documentation for an individual action indicates whether that action returns an encoded message in addition to returning an HTTP code. The message is encoded because the details of the authorization status can constitute privileged information that the user who requested the action should not see. To decode an authorization status message, a user must be granted permissions via an IAM policy to request the *DecodeAuthorizationMessage* (*sts:DecodeAuthorizationMessage*) action.

The decoded message includes the following type of information:

- Whether the request was denied due to an explicit deny or due to the absence of an explicit allow. For more information, see **‘Determining Whether a Request is Allowed or Denied’** in Using IAM .
- The principal who made the request.
- The requested action.
- The requested resource.
- The values of condition keys in the context of the user’s request.

Parameters `encoded_message` (*string*) – The encoded message that was returned with the response.

get_federation_token (*name*, *duration=None*, *policy=None*)

Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. A typical use is in a proxy application that is getting temporary security credentials on behalf of distributed applications inside a corporate network. Because you must call the *GetFederationToken* action using the long- term security credentials of an IAM user, this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based application.

Note: Do not use this call in mobile applications or client-based web applications that directly get temporary security credentials. For those types of applications, use *AssumeRoleWithWebIdentity*.

The *GetFederationToken* action must be called by using the long-term AWS security credentials of the AWS account or an IAM user. Credentials that are created by IAM users are valid for the specified duration, between 900 seconds (15 minutes) and 129600 seconds (36 hours); credentials that are created by using account credentials have a maximum duration of 3600 seconds (1 hour).

The permissions that are granted to the federated user are the intersection of the policy that is passed with the *GetFederationToken* request and policies that are associated with of the entity making the *GetFederationToken* call.

For more information about how permissions work, see **‘Controlling Permissions in Temporary Credentials’** in Using Temporary Security Credentials . For information about using *GetFederationToken* to create temporary security credentials, see **‘Creating Temporary Credentials to Enable Access for Federated Users’** in Using Temporary Security Credentials .

Parameters

- **name** (*string*) – The name of the federated user. The name is used as an identifier for the temporary security credentials (such as *Bob*). For example, you can reference the federated user name in a resource-based policy, such as in an Amazon S3 bucket policy.
- **policy** (*string*) – A policy that specifies the permissions that are granted to the federated user. By default, federated users have no permissions; they do not inherit any from the IAM user. When you specify a policy, the federated user’s permissions are intersection of the specified policy and the IAM user’s policy. If you don’t specify a policy, federated users can only access AWS resources that explicitly allow those federated users in a resource policy, such as in an Amazon S3 bucket policy.

- **duration** (*integer*) – The duration, in seconds, that the session should last. Acceptable durations for federation sessions range from 900 seconds (15 minutes) to 129600 seconds (36 hours), with 43200 seconds (12 hours) as the default. Sessions for AWS account owners are restricted to a maximum of 3600 seconds (one hour). If the duration is longer than one hour, the session for AWS account owners defaults to one hour.

get_session_token (*duration=None, force_new=False, mfa_serial_number=None, mfa_token=None*)

Return a valid session token. Because retrieving new tokens from the Secure Token Service is a fairly heavyweight operation this module caches previously retrieved tokens and returns them when appropriate. Each token is cached with a key consisting of the region name of the STS endpoint concatenated with the requesting user's access id. If there is a token in the cache meeting with this key, the session expiration is checked to make sure it is still valid and if so, the cached token is returned. Otherwise, a new session token is requested from STS and it is placed into the cache and returned.

Parameters

- **duration** (*int*) – The number of seconds the credentials should remain valid.
- **force_new** (*bool*) – If this parameter is True, a new session token will be retrieved from the Secure Token Service regardless of whether there is a valid cached token or not.
- **mfa_serial_number** (*str*) – The serial number of an MFA device. If this is provided and if the mfa_passcode provided is valid, the temporary session token will be authorized with to perform operations requiring the MFA device authentication.
- **mfa_token** (*str*) – The 6 digit token associated with the MFA device.

boto.sts.credentials

class boto.sts.credentials.**AssumedRole** (*connection=None, credentials=None, user=None*)

Variables

- **user** – The assumed role user.
- **credentials** – A Credentials object containing the credentials.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.sts.credentials.**Credentials** (*parent=None*)

Variables

- **access_key** – The AccessKeyID.
- **secret_key** – The SecretAccessKey.
- **session_token** – The session token that must be passed with requests to use the temporary credentials
- **expiration** – The timestamp for when the credentials will expire

endElement (*name, value, connection*)

classmethod **from_json** (*json_doc*)

Create and return a new Session Token based on the contents of a JSON document.

Parameters **json_doc** (*str*) – A string containing a JSON document with a previously saved Credentials object.

is_expired (*time_offset_seconds=0*)

Checks to see if the Session Token is expired or not. By default it will check to see if the Session Token is expired as of the moment the method is called. However, you can supply an optional parameter which is the number of seconds of offset into the future for the check. For example, if you supply a value of 5, this method will return a True if the Session Token will be expired 5 seconds from this moment.

Parameters **time_offset_seconds** (*int*) – The number of seconds into the future to test the Session Token for expiration.

classmethod load (*file_path*)

Create and return a new Session Token based on the contents of a previously saved JSON-format file.

Parameters **file_path** (*str*) – The fully qualified path to the JSON-format file containing the previously saved Session Token information.

save (*file_path*)

Persist a Session Token to a file in JSON format.

Parameters **path** (*str*) – The fully qualified path to the file where the the Session Token data should be written. Any previous data in the file will be overwritten. To help protect the credentials contained in the file, the permissions of the file will be set to readable/writable by owner only.

startElement (*name, attrs, connection*)

to_dict ()

Return a Python dict containing the important information about this Session Token.

class `boto.sts.credentials.DecodeAuthorizationMessage` (*request_id=None, decoded_message=None*)

Variables

- **request_id** – The request ID.
- **decoded_message** – The decoded authorization message (may be JSON).

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.sts.credentials.FederationToken` (*parent=None*)

Variables

- **credentials** – A Credentials object containing the credentials.
- **federated_user_arn** – ARN specifying federated user using credentials.
- **federated_user_id** – The ID of the federated user using credentials.
- **packed_policy_size** – A percentage value indicating the size of the policy in packed form

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.sts.credentials.User` (*arn=None, assume_role_id=None*)

Variables

- **arn** – The arn of the user assuming the role.
- **assume_role_id** – The identifier of the assumed role.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.138.16 SWF

boto.swf

`boto.swf.connect_to_region` (*region_name*, ***kw_params*)

`boto.swf.regions` (***kw_params*)

Get all available regions for the Amazon Simple Workflow service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

boto.swf.layer1

class `boto.swf.layer1.Layer1` (*aws_access_key_id=None*, *aws_secret_access_key=None*,
is_secure=True, *port=None*, *proxy=None*, *proxy_port=None*, *debug=0*, *session_token=None*, *region=None*, *profile_name=None*)

Low-level interface to Simple Workflow Service.

DefaultRegionName = 'us-east-1'

The default region name for Simple Workflow.

ResponseError

alias of `boto.exception.SWFResponseError`

ServiceName = 'com.amazonaws.swf.service.model.SimpleWorkflowService'

The name of the Service

count_closed_workflow_executions (*domain*, *start_latest_date=None*,
start_oldest_date=None, *close_latest_date=None*,
close_oldest_date=None, *close_status=None*,
tag=None, *workflow_id=None*, *workflow_name=None*,
workflow_version=None)

Returns the number of closed workflow executions within the given domain that meet the specified filtering criteria.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow executions to count.
- **start_latest_date** (*timestamp*) – If specified, only workflow executions that meet the start time criteria of the filter are counted.
- **start_oldest_date** (*timestamp*) – If specified, only workflow executions that meet the start time criteria of the filter are counted.
- **close_latest_date** (*timestamp*) – If specified, only workflow executions that meet the close time criteria of the filter are counted.
- **close_oldest_date** (*timestamp*) – If specified, only workflow executions that meet the close time criteria of the filter are counted.
- **close_status** (*string*) – The close status that must match the close status of an execution for it to meet the criteria of this filter. Valid values are:
 - COMPLETED
 - FAILED
 - CANCELED
 - TERMINATED

- CONTINUED_AS_NEW
- TIMED_OUT

- **tag** (*string*) – If specified, only executions that have a tag that matches the filter are counted.
- **workflow_id** (*string*) – If specified, only workflow executions matching the workflow_id are counted.
- **workflow_name** (*string*) – Name of the workflow type to filter on.
- **workflow_version** (*string*) – Version of the workflow type to filter on.

Raises UnknownResourceFault, SWFOperationNotPermittedError

count_open_workflow_executions (*domain, latest_date, oldest_date, tag=None, workflow_id=None, workflow_name=None, workflow_version=None*)

Returns the number of open workflow executions within the given domain that meet the specified filtering criteria.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow executions to count.
- **latest_date** (*timestamp*) – Specifies the latest start or close date and time to return.
- **oldest_date** (*timestamp*) – Specifies the oldest start or close date and time to return.
- **workflow_name** (*string*) – Name of the workflow type to filter on.
- **workflow_version** (*string*) – Version of the workflow type to filter on.
- **tag** (*string*) – If specified, only executions that have a tag that matches the filter are counted.
- **workflow_id** (*string*) – If specified, only workflow executions matching the workflow_id are counted.

Raises UnknownResourceFault, SWFOperationNotPermittedError

count_pending_activity_tasks (*domain, task_list*)

Returns the estimated number of activity tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no activity task was ever scheduled in then 0 will be returned.

Parameters

- **domain** (*string*) – The name of the domain that contains the task list.
- **task_list** (*string*) – The name of the task list.

Raises UnknownResourceFault, SWFOperationNotPermittedError

count_pending_decision_tasks (*domain, task_list*)

Returns the estimated number of decision tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no decision task was ever scheduled in then 0 will be returned.

Parameters

- **domain** (*string*) – The name of the domain that contains the task list.
- **task_list** (*string*) – The name of the task list.

Raises UnknownResourceFault, SWFOperationNotPermittedError

deprecate_activity_type (*domain*, *activity_name*, *activity_version*)

Deprecates the specified activity type. After an activity type has been deprecated, you cannot create new tasks of that activity type. Tasks of this type that were scheduled before the type was deprecated will continue to run.

Parameters

- **domain** (*string*) – The name of the domain in which the activity type is registered.
- **activity_name** (*string*) – The name of this activity.
- **activity_version** (*string*) – The version of this activity.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFOperationNotPermittedError

deprecate_domain (*name*)

Deprecates the specified domain. After a domain has been deprecated it cannot be used to create new workflow executions or register new types. However, you can still use visibility actions on this domain. Deprecating a domain also deprecates all activity and workflow types registered in the domain. Executions that were started before the domain was deprecated will continue to run.

Parameters **name** (*string*) – The name of the domain to deprecate.

Raises UnknownResourceFault, DomainDeprecatedFault, SWFOperationNotPermittedError

deprecate_workflow_type (*domain*, *workflow_name*, *workflow_version*)

Deprecates the specified workflow type. After a workflow type has been deprecated, you cannot create new executions of that type. Executions that were started before the type was deprecated will continue to run. A deprecated workflow type may still be used when calling visibility actions.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow type is registered.
- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFOperationNotPermittedError

describe_activity_type (*domain*, *activity_name*, *activity_version*)

Returns information about the specified activity type. This includes configuration settings provided at registration time as well as other general information about the type.

Parameters

- **domain** (*string*) – The name of the domain in which the activity type is registered.
- **activity_name** (*string*) – The name of this activity.
- **activity_version** (*string*) – The version of this activity.

Raises UnknownResourceFault, SWFOperationNotPermittedError

describe_domain (*name*)

Returns information about the specified domain including description and status.

Parameters **name** (*string*) – The name of the domain to describe.

Raises UnknownResourceFault, SWFOperationNotPermittedError

describe_workflow_execution (*domain*, *run_id*, *workflow_id*)

Returns information about the specified workflow execution including its type and some statistics.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution.
- **run_id** (*string*) – A system generated unique identifier for the workflow execution.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution.

Raises UnknownResourceFault, SWFOperationNotPermittedError

describe_workflow_type (*domain, workflow_name, workflow_version*)

Returns information about the specified workflow type. This includes configuration settings specified when the type was registered and other information such as creation date, current status, etc.

Parameters

- **domain** (*string*) – The name of the domain in which this workflow type is registered.
- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.

Raises UnknownResourceFault, SWFOperationNotPermittedError

get_workflow_execution_history (*domain, run_id, workflow_id, maximum_page_size=None, next_page_token=None, reverse_order=None*)

Returns the history of the specified workflow execution. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the nextPageToken returned by the initial call.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution.
- **run_id** (*string*) – A system generated unique identifier for the workflow execution.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution.
- **maximum_page_size** (*integer*) – Specifies the maximum number of history events returned in one page. The next page in the result is identified by the nextPageToken returned. By default 100 history events are returned in a page but the caller can override this value to a page size smaller than the default. You cannot specify a page size larger than 100.
- **next_page_token** (*string*) – If a nextPageToken is returned, the result has more than one pages. To get the next page, repeat the call and specify the nextPageToken with all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the events in reverse order. By default the results are returned in ascending order of the eventTimeStamp of the events.

Raises UnknownResourceFault, SWFOperationNotPermittedError

json_request (*action, data, object_hook=None*)

This method wraps around make_request() to normalize and serialize the dictionary with request parameters.

Parameters

- **action** (*string*) – Specifies an SWF action.

- **data** (*dict*) – Specifies request parameters associated with the action.

list_activity_types (*domain*, *registration_status*, *name=None*, *maximum_page_size=None*, *next_page_token=None*, *reverse_order=None*)

Returns information about all activities registered in the specified domain that match the specified name and registration status. The result includes information like creation date, current status of the activity, etc. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Parameters

- **domain** (*string*) – The name of the domain in which the activity types have been registered.
- **registration_status** (*string*) – Specifies the registration status of the activity types to list. Valid values are:
 - REGISTERED
 - DEPRECATED
- **name** (*string*) – If specified, only lists the activity types that have this name.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a `NextResultToken` was returned, the results have more than one page. To get the next page of results, repeat the call with the `nextPageToken` and keep all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the activity types.

Raises `SWFOperationNotPermittedError`, `UnknownResourceFault`

list_closed_workflow_executions (*domain*, *start_latest_date=None*, *start_oldest_date=None*, *close_latest_date=None*, *close_oldest_date=None*, *close_status=None*, *tag=None*, *workflow_id=None*, *workflow_name=None*, *workflow_version=None*, *maximum_page_size=None*, *next_page_token=None*, *reverse_order=None*)

Returns the number of closed workflow executions within the given domain that meet the specified filtering criteria.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow executions to count.
- **start_latest_date** (*timestamp*) – If specified, only workflow executions that meet the start time criteria of the filter are counted.
- **start_oldest_date** (*timestamp*) – If specified, only workflow executions that meet the start time criteria of the filter are counted.
- **close_latest_date** (*timestamp*) – If specified, only workflow executions that meet the close time criteria of the filter are counted.
- **close_oldest_date** (*timestamp*) – If specified, only workflow executions that meet the close time criteria of the filter are counted.

- **close_status** (*string*) – The close status that must match the close status of an execution for it to meet the criteria of this filter. Valid values are:
 - COMPLETED
 - FAILED
 - CANCELED
 - TERMINATED
 - CONTINUED_AS_NEW
 - TIMED_OUT
- **tag** (*string*) – If specified, only executions that have a tag that matches the filter are counted.
- **workflow_id** (*string*) – If specified, only workflow executions matching the workflow_id are counted.
- **workflow_name** (*string*) – Name of the workflow type to filter on.
- **workflow_version** (*string*) – Version of the workflow type to filter on.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a NextPageToken was returned, the results are being paginated. To get the next page of results, repeat the call with the returned token and all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in descending order of the start or the close time of the executions.

Raises UnknownResourceFault, SWFOperationNotPermittedError

list_domains (*registration_status*, *maximum_page_size=None*, *next_page_token=None*, *reverse_order=None*)

Returns the list of domains registered in the account. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the nextPageToken returned by the initial call.

Parameters

- **registration_status** (*string*) – Specifies the registration status of the domains to list. Valid Values:
 - REGISTERED
 - DEPRECATED
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a NextPageToken was returned, the result has more than one page. To get the next page of results, repeat the call with the returned token and all other arguments unchanged.

- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the domains.

Raises SWFOperationNotPermittedError

list_open_workflow_executions (*domain*, *oldest_date*, *latest_date=None*, *tag=None*, *workflow_id=None*, *workflow_name=None*, *workflow_version=None*, *maximum_page_size=None*, *next_page_token=None*, *reverse_order=None*)

Returns the list of open workflow executions within the given domain that meet the specified filtering criteria.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow executions to count.
- **latest_date** (*timestamp*) – Specifies the latest start or close date and time to return.
- **oldest_date** (*timestamp*) – Specifies the oldest start or close date and time to return.
- **tag** (*string*) – If specified, only executions that have a tag that matches the filter are counted.
- **workflow_id** (*string*) – If specified, only workflow executions matching the *workflow_id* are counted.
- **workflow_name** (*string*) – Name of the workflow type to filter on.
- **workflow_version** (*string*) – Version of the workflow type to filter on.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a `NextPageToken` was returned, the results are being paginated. To get the next page of results, repeat the call with the returned token and all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in descending order of the start or the close time of the executions.

Raises UnknownResourceFault, SWFOperationNotPermittedError

list_workflow_types (*domain*, *registration_status*, *maximum_page_size=None*, *name=None*, *next_page_token=None*, *reverse_order=None*)

Returns information about workflow types in the specified domain. The results may be split into multiple pages that can be retrieved by making the call repeatedly.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow types have been registered.
- **registration_status** (*string*) – Specifies the registration status of the activity types to list. Valid values are:
 - REGISTERED

– DEPRECATED

- **name** (*string*) – If specified, lists the workflow type with this name.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a NextPageToken was returned, the results are being paginated. To get the next page of results, repeat the call with the returned token and all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the workflow types.

Raises SWFOperationNotPermittedError, UnknownResourceFault

make_request (*action, body=”, object_hook=None*)

Raises SWFResponseError if response status is not 200.

poll_for_activity_task (*domain, task_list, identity=None*)

Used by workers to get an ActivityTask from the specified activity taskList. This initiates a long poll, where the service holds the HTTP connection open and responds as soon as a task becomes available. The maximum time the service holds on to the request before responding is 60 seconds. If no task is available within 60 seconds, the poll will return an empty result. An empty result, in this context, means that an ActivityTask is returned, but that the value of taskToken is an empty string. If a task is returned, the worker should use its type to identify and process it correctly.

Parameters

- **domain** (*string*) – The name of the domain that contains the task lists being polled.
- **task_list** (*string*) – Specifies the task list to poll for activity tasks.
- **identity** (*string*) – Identity of the worker making the request, which is recorded in the ActivityTaskStarted event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

Raises UnknownResourceFault, SWFOperationNotPermittedError

poll_for_decision_task (*domain, task_list, identity=None, maximum_page_size=None, next_page_token=None, reverse_order=None*)

Used by deciders to get a DecisionTask from the specified decision taskList. A decision task may be returned for any open workflow execution that is using the specified task list. The task includes a paginated view of the history of the workflow execution. The decider should use the workflow type and the history to determine how to properly handle the task.

Parameters

- **domain** (*string*) – The name of the domain containing the task lists to poll.
- **task_list** (*string*) – Specifies the task list to poll for decision tasks.
- **identity** (*string*) – Identity of the decider making the request, which is recorded in the DecisionTaskStarted event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.
- **next_page_token** (*string*) – If on a previous call to this method a NextPageToken was returned, the results are being paginated. To get the next

page of results, repeat the call with the returned token and all other arguments unchanged.

- **reverse_order** (*boolean*) – When set to true, returns the events in reverse order. By default the results are returned in ascending order of the eventTimestamp of the events.

Raises UnknownResourceFault, SWFOperationNotPermittedError

record_activity_task_heartbeat (*task_token, details=None*)

Used by activity workers to report to the service that the ActivityTask represented by the specified taskToken is still making progress. The worker can also (optionally) specify details of the progress, for example percent complete, using the details parameter. This action can also be used by the worker as a mechanism to check if cancellation is being requested for the activity task. If a cancellation is being attempted for the specified task, then the boolean cancelRequested flag returned by the service is set to true.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – If specified, contains details about the progress of the task.

Raises UnknownResourceFault, SWFOperationNotPermittedError

register_activity_type (*domain, name, version, task_list=None, default_task_heartbeat_timeout=None, default_task_schedule_to_close_timeout=None, default_task_schedule_to_start_timeout=None, default_task_start_to_close_timeout=None, description=None*)

Registers a new activity type along with its configuration settings in the specified domain.

Parameters

- **domain** (*string*) – The name of the domain in which this activity is to be registered.
- **name** (*string*) – The name of the activity type within the domain.
- **version** (*string*) – The version of the activity type.
- **task_list** (*string*) – If set, specifies the default task list to use for scheduling tasks of this activity type. This default task list is used if a task list is not provided when a task is scheduled through the `schedule_activity_task` Decision.
- **default_task_heartbeat_timeout** (*string*) – If set, specifies the default maximum time before which a worker processing a task of this type must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision. If the activity worker subsequently attempts to record a heartbeat or returns a result, the activity worker receives an `UnknownResource` fault. In this case, Amazon SWF no longer considers the activity task to be valid; the activity worker should clean up the activity task.`no docs`
- **default_task_schedule_to_close_timeout** (*string*) – If set, specifies the default maximum duration for a task of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.`no docs`
- **default_task_schedule_to_start_timeout** (*string*) – If set, specifies the default maximum duration that a task of this activity type can wait

before being assigned to a worker. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.

- **default_task_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration that a worker can take to process tasks of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.
- **description** (*string*) – A textual description of the activity type.

Raises `SWFTypeAlreadyExistsError`, `SWFLimitExceededError`, `UnknownResourceFault`, `SWFOperationNotPermittedError`

register_domain (*name*, *workflow_execution_retention_period_in_days*, *description=None*)

Registers a new domain.

Parameters

- **name** (*string*) – Name of the domain to register. The name must be unique.
- **workflow_execution_retention_period_in_days** (*string*) – Specifies the duration *in days* for which the record (including the history) of workflow executions in this domain should be kept by the service. After the retention period, the workflow execution will not be available in the results of visibility calls. If a duration of `NONE` is specified, the records for workflow executions in this domain are not retained at all.
- **description** (*string*) – Textual description of the domain.

Raises `SWFDomainAlreadyExistsError`, `SWFLimitExceededError`, `SWFOperationNotPermittedError`

register_workflow_type (*domain*, *name*, *version*, *task_list=None*, *default_child_policy=None*, *default_execution_start_to_close_timeout=None*, *default_task_start_to_close_timeout=None*, *description=None*)

Registers a new workflow type and its configuration settings in the specified domain.

Parameters

- **domain** (*string*) – The name of the domain in which to register the workflow type.
- **name** (*string*) – The name of the workflow type.
- **version** (*string*) – The version of the workflow type.
- **task_list** (*list of name, version of tasks*) – If set, specifies the default task list to use for scheduling decision tasks for executions of this workflow type. This default is used only if a task list is not provided when starting the execution through the `StartWorkflowExecution` Action or `StartChildWorkflowExecution` Decision.
- **default_child_policy** (*string*) – If set, specifies the default policy to use for the child workflow executions when a workflow execution of this type is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` Decision. The supported child policies are:
 - `TERMINATE`: the child executions will be terminated.
 - `REQUEST_CANCEL`: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its

history. It is up to the decider to take appropriate actions when it receives an execution history with this event.

– **ABANDON**: no action will be taken. The child executions will continue to run.
no docs

- **default_execution_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration for executions of this workflow type. You can override this default when starting an execution through the StartWorkflowExecution Action or StartChildWorkflowExecution Decision.
- **default_task_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration of decision tasks for this workflow type. This default can be overridden when starting a workflow execution using the StartWorkflowExecution action or the StartChildWorkflowExecution Decision.
- **description** (*string*) – Textual description of the workflow type.

Raises SWFTypeAlreadyExistsError, SWFLimitExceededError, UnknownResourceFault, SWFOperationNotPermittedError

request_cancel_workflow_execution (*domain, workflow_id, run_id=None*)

Records a WorkflowExecutionCancelRequested event in the currently running workflow execution identified by the given domain, workflowId, and runId. This logically requests the cancellation of the workflow execution as a whole. It is up to the decider to take appropriate actions when it receives an execution history with this event.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution to cancel.
- **run_id** (*string*) – The runId of the workflow execution to cancel.
- **workflow_id** (*string*) – The workflowId of the workflow execution to cancel.

Raises UnknownResourceFault, SWFOperationNotPermittedError

respond_activity_task_canceled (*task_token, details=None*)

Used by workers to tell the service that the ActivityTask identified by the taskToken was successfully canceled. Additional details can be optionally provided using the details argument.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – Optional detailed information about the failure.

Raises UnknownResourceFault, SWFOperationNotPermittedError

respond_activity_task_completed (*task_token, result=None*)

Used by workers to tell the service that the ActivityTask identified by the taskToken completed successfully with a result (if provided).

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **result** (*string*) – The result of the activity task. It is a free form string that is implementation specific.

Raises UnknownResourceFault, SWFOperationNotPermittedError

respond_activity_task_failed (*task_token*, *details=None*, *reason=None*)

Used by workers to tell the service that the ActivityTask identified by the taskToken has failed with reason (if specified).

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – Optional detailed information about the failure.
- **reason** (*string*) – Description of the error that may assist in diagnostics.

Raises UnknownResourceFault, SWFOperationNotPermittedError

respond_decision_task_completed (*task_token*, *decisions=None*, *execution_context=None*)

Used by deciders to tell the service that the DecisionTask identified by the taskToken has successfully completed. The decisions argument specifies the list of decisions made while processing the task.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **decisions** (*list*) – The list of decisions (possibly empty) made by the decider while processing this decision task. See the docs for the Decision structure for details.
- **execution_context** (*string*) – User defined context to add to workflow execution.

Raises UnknownResourceFault, SWFOperationNotPermittedError

signal_workflow_execution (*domain*, *signal_name*, *workflow_id*, *input=None*, *run_id=None*)

Records a WorkflowExecutionSignaled event in the workflow execution history and creates a decision task for the workflow execution identified by the given domain, workflowId and runId. The event is recorded with the specified user defined signalName and input (if provided).

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution to signal.
- **signal_name** (*string*) – The name of the signal. This name must be meaningful to the target workflow.
- **workflow_id** (*string*) – The workflowId of the workflow execution to signal.
- **input** (*string*) – Data to attach to the WorkflowExecutionSignaled event in the target workflow execution's history.
- **run_id** (*string*) – The runId of the workflow execution to signal.

Raises UnknownResourceFault, SWFOperationNotPermittedError

start_workflow_execution (*domain*, *workflow_id*, *workflow_name*, *workflow_version*, *task_list=None*, *child_policy=None*, *execution_start_to_close_timeout=None*, *input=None*, *tag_list=None*, *task_start_to_close_timeout=None*)

Starts an execution of the workflow type in the specified domain using the provided workflowId and input data.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow execution is created.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution. You can use this to associate a custom identifier with the

workflow execution. You may specify the same identifier if a workflow execution is logically a restart of a previous execution. You cannot have two open workflow executions with the same workflowId at the same time.

- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.
- **task_list** (*string*) – The task list to use for the decision tasks generated for this workflow execution. This overrides the defaultTaskList specified when registering the workflow type.
- **child_policy** (*string*) – If set, specifies the policy to use for the child workflow executions of this workflow execution if it is terminated, by calling the TerminateWorkflowExecution action explicitly or due to an expired timeout. This policy overrides the default child policy specified when registering the workflow type using RegisterWorkflowType. The supported child policies are:
 - **TERMINATE**: the child executions will be terminated.
 - **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a WorkflowExecutionCancelRequested event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
 - **ABANDON**: no action will be taken. The child executions will continue to run.
- **execution_start_to_close_timeout** (*string*) – The total duration for this workflow execution. This overrides the defaultExecutionStartToCloseTimeout specified when registering the workflow type.
- **input** (*string*) – The input for the workflow execution. This is a free form string which should be meaningful to the workflow you are starting. This input is made available to the new workflow execution in the WorkflowExecutionStarted history event.

task_start_to_close_timeout: Specifies the maximum duration of decision tasks for this workflow execution. This parameter overrides the defaultTaskStartToCloseTimeout specified when registering the workflow type using register_workflow_type.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFWorkflowExecutionAlreadyStartedError, SWFLimitExceededError, SWFOperationNotPermittedError, DefaultUndefinedFault

terminate_workflow_execution (*domain, workflow_id, child_policy=None, details=None, reason=None, run_id=None*)

Records a WorkflowExecutionTerminated event and forces closure of the workflow execution identified by the given domain, runId, and workflowId. The child policy, registered with the workflow type or specified when starting this execution, is applied to any open child workflow executions of this workflow execution.

Parameters

- **domain** (*string*) – The domain of the workflow execution to terminate.
- **workflow_id** (*string*) – The workflowId of the workflow execution to terminate.
- **child_policy** (*string*) – If set, specifies the policy to use for the child workflow executions of the workflow execution being terminated. This policy overrides the child policy specified for the workflow execution at registration time or when starting the execution. The supported child policies are:

- TERMINATE: the child executions will be terminated.
 - REQUEST_CANCEL: a request to cancel will be attempted for each child execution by recording a WorkflowExecutionCancelRequested event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
 - ABANDON: no action will be taken. The child executions will continue to run.
- **details** (*string*) – Optional details for terminating the workflow execution.
 - **reason** (*string*) – An optional descriptive reason for terminating the workflow execution.
 - **run_id** (*string*) – The runId of the workflow execution to terminate.

Raises UnknownResourceFault, SWFOperationNotPermittedError

boto.swf.layer1_decisions

Helper class for creating decision responses.

class boto.swf.layer1_decisions.**Layer1Decisions**

Use this object to build a list of decisions for a decision response. Each method call will add append a new decision. Retrieve the list of decisions from the `_data` attribute.

cancel_timer (*timer_id*)

Cancels a previously started timer and records a TimerCanceled event in the history.

cancel_workflow_executions (*details=None*)

Closes the workflow execution and records a WorkflowExecutionCanceled event in the history.

complete_workflow_execution (*result=None*)

Closes the workflow execution and records a WorkflowExecutionCompleted event in the history

continue_as_new_workflow_execution (*child_policy=None, execution_start_to_close_timeout=None, input=None, tag_list=None, task_list=None, start_to_close_timeout=None, workflow_type_version=None*)

Closes the workflow execution and starts a new workflow execution of the same type using the same workflow id and a unique run Id. A WorkflowExecutionContinuedAsNew event is recorded in the history.

fail_workflow_execution (*reason=None, details=None*)

Closes the workflow execution and records a WorkflowExecutionFailed event in the history.

record_marker (*marker_name, details=None*)

Records a MarkerRecorded event in the history. Markers can be used for adding custom information in the history for instance to let deciders know that they do not need to look at the history beyond the marker event.

request_cancel_activity_task (*activity_id*)

Attempts to cancel a previously scheduled activity task. If the activity task was scheduled but has not been assigned to a worker, then it will be canceled. If the activity task was already assigned to a worker, then the worker will be informed that cancellation has been requested in the response to RecordActivity-TaskHeartbeat.

request_cancel_external_workflow_execution (*workflow_id*, *control=None*,
run_id=None)

Requests that a request be made to cancel the specified external workflow execution and records a RequestCancelExternalWorkflowExecutionInitiated event in the history.

schedule_activity_task (*activity_id*, *activity_type_name*, *activity_type_version*,
task_list=None, *control=None*, *heartbeat_timeout=None*, *schedule_to_close_timeout=None*,
schedule_to_start_timeout=None,
start_to_close_timeout=None, *input=None*)

Schedules an activity task.

Parameters

- **activity_id** (*string*) – The activityId of the type of the activity being scheduled.
- **activity_type_name** (*string*) – The name of the type of the activity being scheduled.
- **activity_type_version** (*string*) – The version of the type of the activity being scheduled.
- **task_list** (*string*) – If set, specifies the name of the task list in which to schedule the activity task. If not specified, the defaultTaskList registered with the activity type will be used. Note: a task list for this activity task must be specified either as a default for the activity type or through this field. If neither this field is set nor a default task list was specified at registration time then a fault will be returned.

signal_external_workflow_execution (*workflow_id*, *signal_name*, *run_id=None*, *control=None*, *input=None*)

Requests a signal to be delivered to the specified external workflow execution and records a SignalExternalWorkflowExecutionInitiated event in the history.

start_child_workflow_execution (*workflow_type_name*, *workflow_type_version*,
workflow_id, *child_policy=None*, *control=None*,
execution_start_to_close_timeout=None, *input=None*,
tag_list=None, *task_list=None*,
task_start_to_close_timeout=None)

Requests that a child workflow execution be started and records a StartChildWorkflowExecutionInitiated event in the history. The child workflow execution is a separate workflow execution with its own history.

start_timer (*start_to_fire_timeout*, *timer_id*, *control=None*)

Starts a timer for this workflow execution and records a TimerStarted event in the history. This timer will fire after the specified delay and record a TimerFired event.

boto.swf.layer2

Object-oriented interface to SWF wrapping boto.swf.layer1.Layer1

class boto.swf.layer2.**ActivityType** (**kwargs)

A versioned activity type.

deprecate ()

Deprecates the specified activity type. After an activity type has been deprecated, you cannot create new tasks of that activity type. Tasks of this type that were scheduled before the type was deprecated will continue to run.

Parameters

- **domain** (*string*) – The name of the domain in which the activity type is registered.

- **activity_name** (*string*) – The name of this activity.
- **activity_version** (*string*) – The version of this activity.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFOperationNotPermittedError

describe ()

Returns information about the specified activity type. This includes configuration settings provided at registration time as well as other general information about the type.

Parameters

- **domain** (*string*) – The name of the domain in which the activity type is registered.
- **activity_name** (*string*) – The name of this activity.
- **activity_version** (*string*) – The version of this activity.

Raises UnknownResourceFault, SWFOperationNotPermittedError

register (***kwargs*)

Registers a new activity type along with its configuration settings in the specified domain.

Parameters

- **domain** (*string*) – The name of the domain in which this activity is to be registered.
- **name** (*string*) – The name of the activity type within the domain.
- **version** (*string*) – The version of the activity type.
- **task_list** (*string*) – If set, specifies the default task list to use for scheduling tasks of this activity type. This default task list is used if a task list is not provided when a task is scheduled through the `schedule_activity_task` Decision.
- **default_task_heartbeat_timeout** (*string*) – If set, specifies the default maximum time before which a worker processing a task of this type must report progress by calling `RecordActivityTaskHeartbeat`. If the timeout is exceeded, the activity task is automatically timed out. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision. If the activity worker subsequently attempts to record a heartbeat or returns a result, the activity worker receives an `UnknownResource` fault. In this case, Amazon SWF no longer considers the activity task to be valid; the activity worker should clean up the activity task.no docs
- **default_task_schedule_to_close_timeout** (*string*) – If set, specifies the default maximum duration for a task of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.no docs
- **default_task_schedule_to_start_timeout** (*string*) – If set, specifies the default maximum duration that a task of this activity type can wait before being assigned to a worker. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.
- **default_task_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration that a worker can take to process tasks of this activity type. This default can be overridden when scheduling an activity task using the `ScheduleActivityTask` Decision.
- **description** (*string*) – A textual description of the activity type.

Raises SWFTypeAlreadyExistsError, SWFLimitExceededError, UnknownResourceFault, SWFOperationNotPermittedError

class boto.swf.layer2.**ActivityWorker** (**kwargs)

Base class for SimpleWorkflow activity workers.

cancel (task_token=None, details=None)

Used by workers to tell the service that the ActivityTask identified by the taskToken was successfully canceled. Additional details can be optionally provided using the details argument.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – Optional detailed information about the failure.

Raises UnknownResourceFault, SWFOperationNotPermittedError

complete (task_token=None, result=None)

Used by workers to tell the service that the ActivityTask identified by the taskToken completed successfully with a result (if provided).

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **result** (*string*) – The result of the activity task. It is a free form string that is implementation specific.

Raises UnknownResourceFault, SWFOperationNotPermittedError

fail (task_token=None, details=None, reason=None)

Used by workers to tell the service that the ActivityTask identified by the taskToken has failed with reason (if specified).

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – Optional detailed information about the failure.
- **reason** (*string*) – Description of the error that may assist in diagnostics.

Raises UnknownResourceFault, SWFOperationNotPermittedError

heartbeat (task_token=None, details=None)

Used by activity workers to report to the service that the ActivityTask represented by the specified taskToken is still making progress. The worker can also (optionally) specify details of the progress, for example percent complete, using the details parameter. This action can also be used by the worker as a mechanism to check if cancellation is being requested for the activity task. If a cancellation is being attempted for the specified task, then the boolean cancelRequested flag returned by the service is set to true.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **details** (*string*) – If specified, contains details about the progress of the task.

Raises UnknownResourceFault, SWFOperationNotPermittedError

poll (**kwargs)

Used by workers to get an ActivityTask from the specified activity taskList. This initiates a long poll, where the service holds the HTTP connection open and responds as soon as a task becomes available. The maximum time the service holds on to the request before responding is 60 seconds. If no task is available within 60 seconds, the poll will return an empty result. An empty result, in this context, means

that an ActivityTask is returned, but that the value of taskToken is an empty string. If a task is returned, the worker should use its type to identify and process it correctly.

Parameters

- **domain** (*string*) – The name of the domain that contains the task lists being polled.
- **task_list** (*string*) – Specifies the task list to poll for activity tasks.
- **identity** (*string*) – Identity of the worker making the request, which is recorded in the ActivityTaskStarted event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.

Raises UnknownResourceFault, SWFOperationNotPermittedError

class boto.swf.layer2.Decider (**kwargs)

Base class for SimpleWorkflow deciders.

complete (*task_token=None, decisions=None, **kwargs*)

Used by deciders to tell the service that the DecisionTask identified by the taskToken has successfully completed. The decisions argument specifies the list of decisions made while processing the task.

Parameters

- **task_token** (*string*) – The taskToken of the ActivityTask.
- **decisions** (*list*) – The list of decisions (possibly empty) made by the decider while processing this decision task. See the docs for the Decision structure for details.
- **execution_context** (*string*) – User defined context to add to workflow execution.

Raises UnknownResourceFault, SWFOperationNotPermittedError

poll (**kwargs)

Used by deciders to get a DecisionTask from the specified decision taskList. A decision task may be returned for any open workflow execution that is using the specified task list. The task includes a paginated view of the history of the workflow execution. The decider should use the workflow type and the history to determine how to properly handle the task.

Parameters

- **domain** (*string*) – The name of the domain containing the task lists to poll.
- **task_list** (*string*) – Specifies the task list to poll for decision tasks.
- **identity** (*string*) – Identity of the decider making the request, which is recorded in the DecisionTaskStarted event in the workflow history. This enables diagnostic tracing when problems arise. The form of this identity is user defined.
- **next_page_token** (*string*) – If on a previous call to this method a NextPageToken was returned, the results are being paginated. To get the next page of results, repeat the call with the returned token and all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the events in reverse order. By default the results are returned in ascending order of the eventTimestamp of the events.

Raises UnknownResourceFault, SWFOperationNotPermittedError

class boto.swf.layer2.Domain (**kwargs)

Simple Workflow Domain.

activities (*status='REGISTERED', **kwargs*)

Returns information about all activities registered in the specified domain that match the specified name and registration status. The result includes information like creation date, current status of the activity, etc. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the `nextPageToken` returned by the initial call.

Parameters

- **domain** (*string*) – The name of the domain in which the activity types have been registered.
- **registration_status** (*string*) – Specifies the registration status of the activity types to list. Valid values are:
 - REGISTERED
 - DEPRECATED
- **name** (*string*) – If specified, only lists the activity types that have this name.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a `NextResultToken` was returned, the results have more than one page. To get the next page of results, repeat the call with the `nextPageToken` and keep all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the activity types.

Raises `SWFOperationNotPermittedError`, `UnknownResourceFault`

count_pending_activity_tasks (*task_list*)

Returns the estimated number of activity tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no activity task was ever scheduled in then 0 will be returned.

Parameters

- **domain** (*string*) – The name of the domain that contains the task list.
- **task_list** (*string*) – The name of the task list.

Raises `UnknownResourceFault`, `SWFOperationNotPermittedError`

count_pending_decision_tasks (*task_list*)

Returns the estimated number of decision tasks in the specified task list. The count returned is an approximation and is not guaranteed to be exact. If you specify a task list that no decision task was ever scheduled in then 0 will be returned.

Parameters

- **domain** (*string*) – The name of the domain that contains the task list.
- **task_list** (*string*) – The name of the task list.

Raises `UnknownResourceFault`, `SWFOperationNotPermittedError`

deprecate ()

Deprecates the specified domain. After a domain has been deprecated it cannot be used to create new workflow executions or register new types. However, you can still use visibility actions on this domain.

Deprecating a domain also deprecates all activity and workflow types registered in the domain. Executions that were started before the domain was deprecated will continue to run.

Parameters `name` (*string*) – The name of the domain to deprecate.

Raises `UnknownResourceFault`, `DomainDeprecatedFault`, `SWFOperationNotPermittedError`

describe ()

Returns information about the specified domain including description and status.

Parameters `name` (*string*) – The name of the domain to describe.

Raises `UnknownResourceFault`, `SWFOperationNotPermittedError`

executions (*closed=False, **kwargs*)

List list open/closed executions.

For a full list of available parameters refer to `boto.swf.layer1.Layer1.list_closed_workflow_executions()` and `boto.swf.layer1.Layer1.list_open_workflow_executions()`

register ()

Registers a new domain.

Parameters

- **name** (*string*) – Name of the domain to register. The name must be unique.
- **workflow_execution_retention_period_in_days** (*string*) – Specifies the duration *in days* for which the record (including the history) of workflow executions in this domain should be kept by the service. After the retention period, the workflow execution will not be available in the results of visibility calls. If a duration of `NONE` is specified, the records for workflow executions in this domain are not retained at all.
- **description** (*string*) – Textual description of the domain.

Raises `SWFDomainAlreadyExistsError`, `SWFLimitExceededError`, `SWFOperationNotPermittedError`

workflows (*status='REGISTERED', **kwargs*)

Returns information about workflow types in the specified domain. The results may be split into multiple pages that can be retrieved by making the call repeatedly.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow types have been registered.
- **registration_status** (*string*) – Specifies the registration status of the activity types to list. Valid values are:
 - `REGISTERED`
 - `DEPRECATED`
- **name** (*string*) – If specified, lists the workflow type with this name.
- **maximum_page_size** (*integer*) – The maximum number of results returned in each page. The default is 100, but the caller can override this value to a page size smaller than the default. You cannot specify a page size greater than 100.
- **next_page_token** (*string*) – If on a previous call to this method a `NextPageToken` was returned, the results are being paginated. To get the next

page of results, repeat the call with the returned token and all other arguments unchanged.

- **reverse_order** (*boolean*) – When set to true, returns the results in reverse order. By default the results are returned in ascending alphabetical order of the name of the workflow types.

Raises SWFOperationNotPermittedError, UnknownResourceFault

class boto.swf.layer2.**WorkflowExecution** (**kwargs)

An instance of a workflow.

describe ()

Returns information about the specified workflow execution including its type and some statistics.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution.
- **run_id** (*string*) – A system generated unique identifier for the workflow execution.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution.

Raises UnknownResourceFault, SWFOperationNotPermittedError

history (**kwargs)

Returns the history of the specified workflow execution. The results may be split into multiple pages. To retrieve subsequent pages, make the call again using the nextPageToken returned by the initial call.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution.
- **run_id** (*string*) – A system generated unique identifier for the workflow execution.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution.
- **maximum_page_size** (*integer*) – Specifies the maximum number of history events returned in one page. The next page in the result is identified by the nextPageToken returned. By default 100 history events are returned in a page but the caller can override this value to a page size smaller than the default. You cannot specify a page size larger than 100.
- **next_page_token** (*string*) – If a nextPageToken is returned, the result has more than one pages. To get the next page, repeat the call and specify the nextPageToken with all other arguments unchanged.
- **reverse_order** (*boolean*) – When set to true, returns the events in reverse order. By default the results are returned in ascending order of the eventTimeStamp of the events.

Raises UnknownResourceFault, SWFOperationNotPermittedError

request_cancel ()

Records a WorkflowExecutionCancelRequested event in the currently running workflow execution identified by the given domain, workflowId, and runId. This logically requests the cancellation of the workflow execution as a whole. It is up to the decider to take appropriate actions when it receives an execution history with this event.

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution to cancel.
- **run_id** (*string*) – The runId of the workflow execution to cancel.
- **workflow_id** (*string*) – The workflowId of the workflow execution to cancel.

Raises UnknownResourceFault, SWFOperationNotPermittedError

signal (*signame*, ***kwargs*)

Records a WorkflowExecutionSignaled event in the workflow execution history and creates a decision task for the workflow execution identified by the given domain, workflowId and runId. The event is recorded with the specified user defined signalName and input (if provided).

Parameters

- **domain** (*string*) – The name of the domain containing the workflow execution to signal.
- **signal_name** (*string*) – The name of the signal. This name must be meaningful to the target workflow.
- **workflow_id** (*string*) – The workflowId of the workflow execution to signal.
- **input** (*string*) – Data to attach to the WorkflowExecutionSignaled event in the target workflow execution’s history.
- **run_id** (*string*) – The runId of the workflow execution to signal.

Raises UnknownResourceFault, SWFOperationNotPermittedError

terminate (***kwargs*)

Records a WorkflowExecutionTerminated event and forces closure of the workflow execution identified by the given domain, runId, and workflowId. The child policy, registered with the workflow type or specified when starting this execution, is applied to any open child workflow executions of this workflow execution.

Parameters

- **domain** (*string*) – The domain of the workflow execution to terminate.
- **workflow_id** (*string*) – The workflowId of the workflow execution to terminate.
- **child_policy** (*string*) – If set, specifies the policy to use for the child workflow executions of the workflow execution being terminated. This policy overrides the child policy specified for the workflow execution at registration time or when starting the execution. The supported child policies are:
 - TERMINATE: the child executions will be terminated.
 - REQUEST_CANCEL: a request to cancel will be attempted for each child execution by recording a WorkflowExecutionCancelRequested event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
 - ABANDON: no action will be taken. The child executions will continue to run.
- **details** (*string*) – Optional details for terminating the workflow execution.

- **reason** (*string*) – An optional descriptive reason for terminating the workflow execution.
- **run_id** (*string*) – The runId of the workflow execution to terminate.

Raises UnknownResourceFault, SWFOperationNotPermittedError

class boto.swf.layer2.**WorkflowType** (**kwargs)

A versioned workflow type.

deprecate ()

Deprecates the specified workflow type. After a workflow type has been deprecated, you cannot create new executions of that type. Executions that were started before the type was deprecated will continue to run. A deprecated workflow type may still be used when calling visibility actions.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow type is registered.
- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFOperationNotPermittedError

describe ()

Returns information about the specified workflow type. This includes configuration settings specified when the type was registered and other information such as creation date, current status, etc.

Parameters

- **domain** (*string*) – The name of the domain in which this workflow type is registered.
- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.

Raises UnknownResourceFault, SWFOperationNotPermittedError

register (**kwargs)

Registers a new workflow type and its configuration settings in the specified domain.

Parameters

- **domain** (*string*) – The name of the domain in which to register the workflow type.
- **name** (*string*) – The name of the workflow type.
- **version** (*string*) – The version of the workflow type.
- **task_list** (*list of name, version of tasks*) – If set, specifies the default task list to use for scheduling decision tasks for executions of this workflow type. This default is used only if a task list is not provided when starting the execution through the StartWorkflowExecution Action or StartChildWorkflowExecution Decision.
- **default_child_policy** (*string*) – If set, specifies the default policy to use for the child workflow executions when a workflow execution of this type is terminated, by calling the TerminateWorkflowExecution action explicitly or due to an expired timeout. This default can be overridden when starting a workflow execution using the StartWorkflowExecution action or the StartChildWorkflowExecution Decision. The supported child policies are:
 - **TERMINATE**: the child executions will be terminated.

- **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
- **ABANDON**: no action will be taken. The child executions will continue to run.
no docs
- **default_execution_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration for executions of this workflow type. You can override this default when starting an execution through the `StartWorkflowExecution` Action or `StartChildWorkflowExecution` Decision.
- **default_task_start_to_close_timeout** (*string*) – If set, specifies the default maximum duration of decision tasks for this workflow type. This default can be overridden when starting a workflow execution using the `StartWorkflowExecution` action or the `StartChildWorkflowExecution` Decision.
- **description** (*string*) – Textual description of the workflow type.

Raises `SWFTypeAlreadyExistsError`, `SWFLimitExceededError`, `UnknownResourceFault`, `SWFOperationNotPermittedError`

start (**kwargs)

Starts an execution of the workflow type in the specified domain using the provided `workflowId` and input data.

Parameters

- **domain** (*string*) – The name of the domain in which the workflow execution is created.
- **workflow_id** (*string*) – The user defined identifier associated with the workflow execution. You can use this to associate a custom identifier with the workflow execution. You may specify the same identifier if a workflow execution is logically a restart of a previous execution. You cannot have two open workflow executions with the same `workflowId` at the same time.
- **workflow_name** (*string*) – The name of the workflow type.
- **workflow_version** (*string*) – The version of the workflow type.
- **task_list** (*string*) – The task list to use for the decision tasks generated for this workflow execution. This overrides the `defaultTaskList` specified when registering the workflow type.
- **child_policy** (*string*) – If set, specifies the policy to use for the child workflow executions of this workflow execution if it is terminated, by calling the `TerminateWorkflowExecution` action explicitly or due to an expired timeout. This policy overrides the default child policy specified when registering the workflow type using `RegisterWorkflowType`. The supported child policies are:
 - **TERMINATE**: the child executions will be terminated.
 - **REQUEST_CANCEL**: a request to cancel will be attempted for each child execution by recording a `WorkflowExecutionCancelRequested` event in its history. It is up to the decider to take appropriate actions when it receives an execution history with this event.
 - **ABANDON**: no action will be taken. The child executions will continue to run.

- **execution_start_to_close_timeout** (*string*) – The total duration for this workflow execution. This overrides the defaultExecutionStartToCloseTimeout specified when registering the workflow type.
- **input** (*string*) – The input for the workflow execution. This is a free form string which should be meaningful to the workflow you are starting. This input is made available to the new workflow execution in the WorkflowExecutionStarted history event.

task_start_to_close_timeout: Specifies the maximum duration of decision tasks for this workflow execution. This parameter overrides the defaultTaskStartToCloseTimeout specified when registering the workflow type using register_workflow_type.

Raises UnknownResourceFault, TypeDeprecatedFault, SWFWorkflowExecutionAlreadyStartedError, SWFLimitExceededError, SWFOperationNotPermittedError, DefaultUndefinedFault

`boto.swf.layer2.set_default_credentials(aws_access_key_id, aws_secret_access_key)`
Set default credentials.

4.138.17 VPC

boto.vpc

Represents a connection to the EC2 service.

```
class boto.vpc.VPCConnection(aws_access_key_id=None, aws_secret_access_key=None,
                             is_secure=True, host=None, port=None, proxy=None,
                             proxy_port=None, proxy_user=None, proxy_pass=None, debug=0,
                             https_connection_factory=None, region=None, path='/',
                             api_version=None, security_token=None, validate_certs=True,
                             profile_name=None)
```

Init method to create a new connection to EC2.

```
accept_vpc_peering_connection(vpc_peering_connection_id, dry_run=False)
```

Accepts a VPC peering connection request. The VPC peering connection must be in the pending-acceptance state.

Parameters `vpc_peering_connection_id` (*str*) – The ID of the VPC peering connection.

Return type Accepted VpcPeeringConnection

Returns A `boto.vpc.vpc_peering_connection.VpcPeeringConnection` object

```
associate_dhcp_options(dhcp_options_id, vpc_id, dry_run=False)
```

Associate a set of Dhcpc Options with a VPC.

Parameters

- **dhcp_options_id** (*str*) – The ID of the Dhcpc Options
- **vpc_id** (*str*) – The ID of the VPC.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

```
associate_network_acl(network_acl_id, subnet_id)
```

Associates a network acl with a specific subnet.

Parameters

- **network_acl_id** (*str*) – The ID of the network ACL to associate.
- **subnet_id** (*str*) – The ID of the subnet to associate with.

Return type *str***Returns** The ID of the association created**associate_route_table** (*route_table_id, subnet_id, dry_run=False*)

Associates a route table with a specific subnet.

Parameters

- **route_table_id** (*str*) – The ID of the route table to associate.
- **subnet_id** (*str*) – The ID of the subnet to associate with.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *str***Returns** The ID of the association created**attach_classic_link_vpc** (*vpc_id, instance_id, groups, dry_run=False*)

Links an EC2-Classic instance to a ClassicLink-enabled VPC through one or more of the VPC's security groups. You cannot link an EC2-Classic instance to more than one VPC at a time. You can only link an instance that's in the running state. An instance is automatically unlinked from a VPC when it's stopped. You can link it to the VPC again when you restart it.

After you've linked an instance, you cannot change the VPC security groups that are associated with it. To change the security groups, you must first unlink the instance, and then link it again.

Linking your instance to a VPC is sometimes referred to as attaching your instance.

Parameters

- **vpc_id** (*str*) – The ID of a ClassicLink-enabled VPC.
- **instance_id** – The ID of a ClassicLink-enabled VPC.
- **groups** – The ID of one or more of the VPC's security groups. You cannot specify security groups from a different VPC. The members of the list can be *boto.ec2.securitygroup.SecurityGroup* objects or strings of the id's of the security groups.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Security groups *list***Return type** *bool***Returns** True if successful**attach_internet_gateway** (*internet_gateway_id, vpc_id, dry_run=False*)

Attach an internet gateway to a specific VPC.

Parameters

- **internet_gateway_id** (*str*) – The ID of the internet gateway to attach.
- **vpc_id** (*str*) – The ID of the VPC to attach to.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *Bool***Returns** True if successful

attach_vpn_gateway (*vpn_gateway_id*, *vpc_id*, *dry_run=False*)

Attaches a VPN gateway to a VPC.

Parameters

- **vpn_gateway_id** (*str*) – The ID of the vpn_gateway to attach
- **vpc_id** (*str*) – The ID of the VPC you want to attach the gateway to.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type An attachment

Returns a *boto.vpc.vpngateway.Attachment*

create_customer_gateway (*type*, *ip_address*, *bgp_asn*, *dry_run=False*)

Create a new Customer Gateway

Parameters

- **type** (*str*) – Type of VPN Connection. Only valid value currently is 'ipsec.1'
- **ip_address** (*str*) – Internet-routable IP address for customer's gateway. Must be a static address.
- **bgp_asn** (*int*) – Customer gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN)
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created CustomerGateway

Returns A *boto.vpc.customergateway.CustomerGateway* object

create_dhcp_options (*domain_name=None*, *domain_name_servers=None*, *ntp_servers=None*, *netbios_name_servers=None*, *netbios_node_type=None*, *dry_run=False*)

Create a new DhcpOption

This corresponds to <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/ApiReference-query-CreateDhcpOptions.html>

Parameters

- **domain_name** (*str*) – A domain name of your choice (for example, example.com)
- **domain_name_servers** (*list of strings*) – The IP address of a domain name server. You can specify up to four addresses.
- **ntp_servers** (*list of strings*) – The IP address of a Network Time Protocol (NTP) server. You can specify up to four addresses.
- **netbios_name_servers** (*list of strings*) – The IP address of a NetBIOS name server. You can specify up to four addresses.
- **netbios_node_type** (*str*) – The NetBIOS node type (1, 2, 4, or 8). For more information about the values, see RFC 2132. We recommend you only use 2 at this time (broadcast and multicast are currently not supported).
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created DhcpOption

Returns A *boto.vpc.customergateway.DhcpOption* object

create_internet_gateway (*dry_run=False*)

Creates an internet gateway for VPC.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type Newly created internet gateway.

Returns *boto.vpc.internetgateway.InternetGateway*

create_network_acl (*vpc_id*)

Creates a new network ACL.

Parameters **vpc_id** (*str*) – The VPC ID to associate this network ACL with.

Return type The newly created network ACL

Returns A *boto.vpc.networkacl.NetworkAcl* object

create_network_acl_entry (*network_acl_id*, *rule_number*, *protocol*, *rule_action*,
cidr_block, *egress=**None*, *icmp_code=**None*, *icmp_type=**None*,
*port_range_from=**None*, *port_range_to=**None*)

Creates a new network ACL entry in a network ACL within a VPC.

Parameters

- **network_acl_id** (*str*) – The ID of the network ACL for this network ACL entry.
- **rule_number** (*int*) – The rule number to assign to the entry (for example, 100).
- **protocol** (*int*) – Valid values: -1 or a protocol number (<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>)

Parameters

- **rule_action** (*str*) – Indicates whether to allow or deny traffic that matches the rule.
- **cidr_block** (*str*) – The CIDR range to allow or deny, in CIDR notation (for example,

172.16.0.0/24).

Parameters **egress** (*bool*) – Indicates whether this rule applies to egress traffic from the subnet (true) or ingress traffic to the subnet (false).

Parameters

- **icmp_type** (*int*) – For the ICMP protocol, the ICMP type. You can use -1 to specify all ICMP types.
- **icmp_code** (*int*) – For the ICMP protocol, the ICMP code. You can use -1 to specify all ICMP codes for the given ICMP type.

Parameters

- **port_range_from** (*int*) – The first port in the range.
- **port_range_to** (*int*) – The last port in the range.

Return type *bool*

Returns True if successful

create_route (*route_table_id*, *destination_cidr_block*, *gateway_id=**None*, *instance_id=**None*, *interface_id=**None*, *vpc_peering_connection_id=**None*, *dry_run=**False*)

Creates a new route in the route table within a VPC. The route's target can be either a gateway attached to the VPC or a NAT instance in the VPC.

Parameters

- **route_table_id** (*str*) – The ID of the route table for the route.

- **destination_cidr_block** (*str*) – The CIDR address block used for the destination match.
- **gateway_id** (*str*) – The ID of the gateway attached to your VPC.
- **instance_id** (*str*) – The ID of a NAT instance in your VPC.
- **interface_id** (*str*) – Allows routing to network interface attachments.
- **vpc_peering_connection_id** (*str*) – Allows routing to VPC peering connection.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *bool*

Returns True if successful

create_route_table (*vpc_id*, *dry_run=False*)

Creates a new route table.

Parameters

- **vpc_id** (*str*) – The VPC ID to associate this route table with.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created route table

Returns A *boto.vpc.routetable.RouteTable* object

create_subnet (*vpc_id*, *cidr_block*, *availability_zone=None*, *dry_run=False*)

Create a new Subnet

Parameters

- **vpc_id** (*str*) – The ID of the VPC where you want to create the subnet.
- **cidr_block** (*str*) – The CIDR block you want the subnet to cover.
- **availability_zone** (*str*) – The AZ you want the subnet in
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created Subnet

Returns A *boto.vpc.customergateway.Subnet* object

create_vpc (*cidr_block*, *instance_tenancy=None*, *dry_run=False*)

Create a new Virtual Private Cloud.

Parameters

- **cidr_block** (*str*) – A valid CIDR block
- **instance_tenancy** (*str*) – The supported tenancy options for instances launched into the VPC. Valid values are ‘default’ and ‘dedicated’.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created VPC

Returns A *boto.vpc.vpc.VPC* object

create_vpc_peering_connection (*vpc_id*, *peer_vpc_id*, *peer_owner_id=None*, *dry_run=False*)

Create a new VPN Peering connection.

Parameters

- **vpc_id** (*str*) – The ID of the requester VPC.

- **vpc_peer_id** – The ID of the VPC with which you are creating the peering connection.
- **peer_owner_id** (*str*) – The AWS account ID of the owner of the peer VPC.

Return type The newly created VpcPeeringConnection

Returns A `boto.vpc.vpc_peering_connection.VpcPeeringConnection` object

create_vpn_connection (*type, customer_gateway_id, vpn_gateway_id, static_routes_only=None, dry_run=False*)

Create a new VPN Connection.

Parameters

- **type** (*str*) – The type of VPN Connection. Currently only ‘ipsec.1’ is supported
- **customer_gateway_id** (*str*) – The ID of the customer gateway.
- **vpn_gateway_id** (*str*) – The ID of the VPN gateway.
- **static_routes_only** (*bool*) – Indicates whether the VPN connection requires static routes. If you are creating a VPN connection for a device that does not support BGP, you must specify true.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created VpnConnection

Returns A `boto.vpc.vpnconnection.VpnConnection` object

create_vpn_connection_route (*destination_cidr_block, vpn_connection_id, dry_run=False*)

Creates a new static route associated with a VPN connection between an existing virtual private gateway and a VPN customer gateway. The static route allows traffic to be routed from the virtual private gateway to the VPN customer gateway.

Parameters

- **destination_cidr_block** (*str*) – The CIDR block associated with the local subnet of the customer data center.
- **vpn_connection_id** (*str*) – The ID of the VPN connection.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

create_vpn_gateway (*type, availability_zone=None, dry_run=False*)

Create a new Vpn Gateway

Parameters

- **type** (*str*) – Type of VPN Connection. Only valid value currently is ‘ipsec.1’
- **availability_zone** (*str*) – The Availability Zone where you want the VPN gateway.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type The newly created VpnGateway

Returns A `boto.vpc.vpngateway.VpnGateway` object

delete_customer_gateway (*customer_gateway_id, dry_run=False*)

Delete a Customer Gateway.

Parameters

- **customer_gateway_id** (*str*) – The ID of the customer_gateway to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_dhcp_options (*dhcp_options_id*, *dry_run=False*)

Delete a DHCP Options

Parameters

- **dhcp_options_id** (*str*) – The ID of the DHCP Options to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_internet_gateway (*internet_gateway_id*, *dry_run=False*)

Deletes an internet gateway from the VPC.

Parameters

- **internet_gateway_id** (*str*) – The ID of the internet gateway to delete.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `Bool`

Returns True if successful

delete_network_acl (*network_acl_id*)

Delete a network ACL

Parameters **network_acl_id** (*str*) – The ID of the network_acl to delete.

Return type `bool`

Returns True if successful

delete_network_acl_entry (*network_acl_id*, *rule_number*, *egress=None*)

Deletes a network ACL entry from a network ACL within a VPC.

Parameters

- **network_acl_id** (*str*) – The ID of the network ACL with the network ACL entry.
- **rule_number** (*int*) – The rule number for the entry to delete.
- **egress** (*bool*) – Specifies whether the rule to delete is an egress rule (true) or ingress rule (false).

Return type `bool`

Returns True if successful

delete_route (*route_table_id*, *destination_cidr_block*, *dry_run=False*)

Deletes a route from a route table within a VPC.

Parameters

- **route_table_id** (*str*) – The ID of the route table with the route.
- **destination_cidr_block** (*str*) – The CIDR address block used for destination match.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_route_table (*route_table_id*, *dry_run=False*)

Delete a route table.

Parameters

- **route_table_id** (*str*) – The ID of the route table to delete.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_subnet (*subnet_id*, *dry_run=False*)

Delete a subnet.

Parameters

- **subnet_id** (*str*) – The ID of the subnet to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_vpc (*vpc_id*, *dry_run=False*)

Delete a Virtual Private Cloud.

Parameters

- **vpc_id** (*str*) – The ID of the vpc to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_vpc_peering_connection (*vpc_peering_connection_id*, *dry_run=False*)

Deletes a VPC peering connection. Either the owner of the requester VPC or the owner of the peer VPC can delete the VPC peering connection if it's in the active state. The owner of the requester VPC can delete a VPC peering connection in the pending-acceptance state.

Parameters **vpc_peering_connection_id** (*str*) – The ID of the VPC peering connection.

Return type `bool`

Returns True if successful

delete_vpn_connection (*vpn_connection_id*, *dry_run=False*)

Delete a VPN Connection.

Parameters

- **vpn_connection_id** (*str*) – The ID of the vpn_connection to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_vpn_connection_route (*destination_cidr_block*, *vpn_connection_id*, *dry_run=False*)

Deletes a static route associated with a VPN connection between an existing virtual private gateway and

a VPN customer gateway. The static route allows traffic to be routed from the virtual private gateway to the VPN customer gateway.

Parameters

- **destination_cidr_block** (*str*) – The CIDR block associated with the local subnet of the customer data center.
- **vpn_connection_id** (*str*) – The ID of the VPN connection.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

delete_vpn_gateway (*vpn_gateway_id*, *dry_run=False*)

Delete a Vpn Gateway.

Parameters

- **vpn_gateway_id** (*str*) – The ID of the vpn_gateway to be deleted.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

detach_classic_link_vpc (*vpc_id*, *instance_id*, *dry_run=False*)

Unlinks a linked EC2-Classic instance from a VPC. After the instance has been unlinked, the VPC security groups are no longer associated with it. An instance is automatically unlinked from a VPC when it's stopped.

Parameters

- **vpc_id** (*str*) – The ID of the instance to unlink from the VPC.
- **instance_id** – The ID of the VPC to which the instance is linked.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

detach_internet_gateway (*internet_gateway_id*, *vpc_id*, *dry_run=False*)

Detach an internet gateway from a specific VPC.

Parameters

- **internet_gateway_id** (*str*) – The ID of the internet gateway to detach.
- **vpc_id** (*str*) – The ID of the VPC to attach to.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `Bool`

Returns True if successful

detach_vpn_gateway (*vpn_gateway_id*, *vpc_id*, *dry_run=False*)

Detaches a VPN gateway from a VPC.

Parameters

- **vpn_gateway_id** (*str*) – The ID of the vpn_gateway to detach
- **vpc_id** (*str*) – The ID of the VPC you want to detach the gateway from.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

disable_vgw_route_propagation (*route_table_id, gateway_id, dry_run=False*)

Disables a virtual private gateway (VGW) from propagating routes to the routing tables of an Amazon VPC.

Parameters

- **route_table_id** (*str*) – The ID of the routing table.
- **gateway_id** (*str*) – The ID of the virtual private gateway.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

disable_vpc_classic_link (*vpc_id, dry_run=False*)

Disables ClassicLink for a VPC. You cannot disable ClassicLink for a VPC that has EC2-Classic instances linked to it.

Parameters

- **vpc_id** (*str*) – The ID of the VPC.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

disassociate_network_acl (*subnet_id, vpc_id=None*)

Figures out what the default ACL is for the VPC, and associates current network ACL with the default.

Parameters

- **subnet_id** (*str*) – The ID of the subnet to which the ACL belongs.
- **vpc_id** (*str*) – The ID of the VPC to which the ACL/subnet belongs. Queries EC2 if omitted.

Return type `str`

Returns The ID of the association created

disassociate_route_table (*association_id, dry_run=False*)

Removes an association from a route table. This will cause all subnets that would've used this association to now use the main routing association instead.

Parameters

- **association_id** (*str*) – The ID of the association to disassociate.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

enable_vgw_route_propagation (*route_table_id, gateway_id, dry_run=False*)

Enables a virtual private gateway (VGW) to propagate routes to the routing tables of an Amazon VPC.

Parameters

- **route_table_id** (*str*) – The ID of the routing table.
- **gateway_id** (*str*) – The ID of the virtual private gateway.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

enable_vpc_classic_link (*vpc_id*, *dry_run=False*)

Enables a VPC for ClassicLink. You can then link EC2-Classic instances to your ClassicLink-enabled VPC to allow communication over private IP addresses. You cannot enable your VPC for ClassicLink if any of your VPC's route tables have existing routes for address ranges within the 10.0.0.0/8 IP address range, excluding local routes for VPCs in the 10.0.0.0/16 and 10.1.0.0/16 IP address ranges.

Parameters

- **vpc_id** (*str*) – The ID of the VPC.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

get_all_classic_link_vpcs (*vpc_ids=None*, *filters=None*, *dry_run=False*)

Describes the ClassicLink status of one or more VPCs.

Parameters

- **vpc_ids** (*list*) – A list of strings with the desired VPC ID's
- **dry_run** (*bool*) – Set to True if the operation should not actually run.
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value.

Return type `list`

Returns A list of `boto.vpc.vpc.VPC`

get_all_customer_gateways (*customer_gateway_ids=None*, *filters=None*, *dry_run=False*)

Retrieve information about your CustomerGateways. You can filter results to return information only about those CustomerGateways that match your search parameters. Otherwise, all CustomerGateways associated with your account are returned.

Parameters

- **customer_gateway_ids** (*list*) – A list of strings with the desired CustomerGateway ID's.
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value. Possible filter keys are:
 - *state*, the state of the CustomerGateway (pending,available,deleting,deleted)
 - *type*, the type of customer gateway (ipsec.1)
 - *ipAddress* the IP address of customer gateway's internet-routable external interface
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `list`

Returns A list of `boto.vpc.customergateway.CustomerGateway`

get_all_dhcp_options (*dhcp_options_ids=None*, *filters=None*, *dry_run=False*)

Retrieve information about your DhcpOptions.

Parameters

- **dhcp_options_ids** (*list*) – A list of strings with the desired DhcpOption ID's
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpc.dhcpoptions.DhcpOptions*

get_all_internet_gateways (*internet_gateway_ids=None, filters=None, dry_run=False*)

Get a list of internet gateways. You can filter results to return information about only those gateways that you're interested in.

Parameters

- **internet_gateway_ids** (*list*) – A list of strings with the desired gateway IDs.
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

get_all_network_acls (*network_acl_ids=None, filters=None*)

Retrieve information about your network acls. You can filter results to return information only about those network acls that match your search parameters. Otherwise, all network acls associated with your account are returned.

Parameters

- **network_acl_ids** (*list*) – A list of strings with the desired network ACL IDs.
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value.

Return type *list*

Returns A list of *boto.vpc.networkacl.NetworkAcl*

get_all_route_tables (*route_table_ids=None, filters=None, dry_run=False*)

Retrieve information about your routing tables. You can filter results to return information only about those route tables that match your search parameters. Otherwise, all route tables associated with your account are returned.

Parameters

- **route_table_ids** (*list*) – A list of strings with the desired route table IDs.
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpc.routetable.RouteTable*

get_all_subnets (*subnet_ids=None, filters=None, dry_run=False*)

Retrieve information about your Subnets. You can filter results to return information only about those Subnets that match your search parameters. Otherwise, all Subnets associated with your account are returned.

Parameters

- **subnet_ids** (*list*) – A list of strings with the desired Subnet ID’s
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value. Possible filter keys are:
 - *state*, a list of states of the Subnet (pending,available)
 - *vpcId*, a list of IDs of the VPC that the subnet is in.
 - *cidrBlock*, a list of CIDR blocks of the subnet
 - *availabilityZone*, list of the Availability Zones the subnet is in.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpc.subnet.Subnet*

get_all_vpc_peering_connections (*vpc_peering_connection_ids=None, filters=None, dry_run=False*)

Retrieve information about your VPC peering connections. You can filter results to return information only about those VPC peering connections that match your search parameters. Otherwise, all VPC peering connections associated with your account are returned.

Parameters

- **vpc_peering_connection_ids** (*list*) – A list of strings with the desired VPC peering connection ID’s
- **filters** (*list of tuples*) – A list of tuples containing filters. Each tuple consists of a filter key and a filter value. Possible filter keys are:
 - *accepter-vpc-info.cidr-block* - The CIDR block of the peer VPC.
 - *accepter-vpc-info.owner-id* - **The AWS account ID of the owner** of the peer VPC.
 - *accepter-vpc-info.vpc-id* - The ID of the peer VPC.
 - *expiration-time* - **The expiration date and time for the VPC** peering connection.
 - *requester-vpc-info.cidr-block* - **The CIDR block of the** requester’s VPC.
 - *requester-vpc-info.owner-id* - **The AWS account ID of the** owner of the requester VPC.
 - *requester-vpc-info.vpc-id* - The ID of the requester VPC.
 - *status-code* - The status of the VPC peering connection.
 - *status-message* - **A message that provides more information** about the status of the VPC peering connection, if applicable.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpc.vpc.VPC*

get_all_vpcs (*vpc_ids=None, filters=None, dry_run=False*)

Retrieve information about your VPCs. You can filter results to return information only about those VPCs that match your search parameters. Otherwise, all VPCs associated with your account are returned.

Parameters

- **vpc_ids** (*list*) – A list of strings with the desired VPC ID’s

- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value. Possible filter keys are:
 - *state* - a list of states of the VPC (pending or available)
 - *cidrBlock* - a list CIDR blocks of the VPC
 - *dhcpOptionsId* - a list of IDs of a set of DHCP options
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpc.vpc.VPC*

get_all_vpn_connections (*vpn_connection_ids=None, filters=None, dry_run=False*)

Retrieve information about your VPN_CONNECTIONs. You can filter results to return information only about those VPN_CONNECTIONs that match your search parameters. Otherwise, all VPN_CONNECTIONs associated with your account are returned.

Parameters

- **vpn_connection_ids** (*list*) – A list of strings with the desired VPN_CONNECTION ID's
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value. Possible filter keys are:
 - *state*, a list of states of the VPN_CONNECTION pending,available,deleting,deleted
 - *type*, a list of types of connection, currently 'ipsec.1'
 - *customerGatewayId*, a list of IDs of the customer gateway associated with the VPN
 - *vpnGatewayId*, a list of IDs of the VPN gateway associated with the VPN connection
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of *boto.vpn_connection.vpnconnection.VpnConnection*

get_all_vpn_gateways (*vpn_gateway_ids=None, filters=None, dry_run=False*)

Retrieve information about your VpnGateways. You can filter results to return information only about those VpnGateways that match your search parameters. Otherwise, all VpnGateways associated with your account are returned.

Parameters

- **vpn_gateway_ids** (*list*) – A list of strings with the desired VpnGateway ID's
- **filters** (*list of tuples or dict*) – A list of tuples or dict containing filters. Each tuple or dict item consists of a filter key and a filter value. Possible filter keys are:
 - *state*, a list of states of the VpnGateway (pending,available,deleting,deleted)
 - *type*, a list types of customer gateway (ipsec.1)
 - *availabilityZone*, a list of Availability zones the VPN gateway is in.

- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *list*

Returns A list of `boto.vpc.customergateway.VpnGateway`

modify_vpc_attribute (*vpc_id*, *enable_dns_support=None*, *enable_dns_hostnames=None*, *dry_run=False*)

Modifies the specified attribute of the specified VPC. You can only modify one attribute at a time.

Parameters

- **vpc_id** (*str*) – The ID of the vpc to be deleted.
- **enable_dns_support** (*bool*) – Specifies whether the DNS server provided by Amazon is enabled for the VPC.
- **enable_dns_hostnames** (*bool*) – Specifies whether DNS hostnames are provided for the instances launched in this VPC. You can only set this attribute to `true` if `EnableDnsSupport` is also `true`.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

reject_vpc_peering_connection (*vpc_peering_connection_id*, *dry_run=False*)

Rejects a VPC peering connection request. The VPC peering connection must be in the pending-acceptance state.

Parameters **vpc_peering_connection_id** (*str*) – The ID of the VPC peering connection.

Return type *bool*

Returns True if successful

replace_network_acl_entry (*network_acl_id*, *rule_number*, *protocol*, *rule_action*, *cidr_block*, *egress=None*, *icmp_code=None*, *icmp_type=None*, *port_range_from=None*, *port_range_to=None*)

Creates a new network ACL entry in a network ACL within a VPC.

Parameters

- **network_acl_id** (*str*) – The ID of the network ACL for the id you want to replace
- **rule_number** (*int*) – The rule number that you want to replace (for example, 100).
- **protocol** (*int*) – Valid values: -1 or a protocol number (<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>)

Parameters

- **rule_action** (*str*) – Indicates whether to allow or deny traffic that matches the rule.
- **cidr_block** (*str*) – The CIDR range to allow or deny, in CIDR notation (for example, 172.16.0.0/24).

Parameters **egress** (*bool*) – Indicates whether this rule applies to egress traffic from the subnet (`true`) or ingress traffic to the subnet (`false`).

Parameters

- **icmp_type** (*int*) – For the ICMP protocol, the ICMP type. You can use -1 to specify all ICMP types.

- **icmp_code** (*int*) – For the ICMP protocol, the ICMP code. You can use -1 to specify all ICMP codes for the given ICMP type.

Parameters

- **port_range_from** (*int*) – The first port in the range.
- **port_range_to** (*int*) – The last port in the range.

Return type `bool`**Returns** True if successful

replace_route (*route_table_id, destination_cidr_block, gateway_id=None, instance_id=None, interface_id=None, vpc_peering_connection_id=None, dry_run=False*)

Replaces an existing route within a route table in a VPC.

Parameters

- **route_table_id** (*str*) – The ID of the route table for the route.
- **destination_cidr_block** (*str*) – The CIDR address block used for the destination match.
- **gateway_id** (*str*) – The ID of the gateway attached to your VPC.
- **instance_id** (*str*) – The ID of a NAT instance in your VPC.
- **interface_id** (*str*) – Allows routing to network interface attachments.
- **vpc_peering_connection_id** (*str*) – Allows routing to VPC peering connection.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`**Returns** True if successful

replace_route_table_association (*association_id, route_table_id, dry_run=False*)

Replaces a route association with a new route table. This can be used to replace the ‘main’ route table by using the main route table association instead of the more common subnet type association.

NOTE: It may be better to use `replace_route_table_association_with_assoc` instead of this function; this function does not return the new association ID. This function is retained for backwards compatibility.

Parameters

- **association_id** (*str*) – The ID of the existing association to replace.
- **route_table_id** (*str*) – The route table to ID to be used in the association.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`**Returns** True if successful

replace_route_table_association_with_assoc (*association_id, route_table_id, dry_run=False*)

Replaces a route association with a new route table. This can be used to replace the ‘main’ route table by using the main route table association instead of the more common subnet type association. Returns the new association ID.

Parameters

- **association_id** (*str*) – The ID of the existing association to replace.
- **route_table_id** (*str*) – The route table to ID to be used in the association.

- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type *str*

Returns New association ID

`boto.vpc.connect_to_region` (*region_name*, ***kw_params*)

Given a valid region name, return a `boto.vpc.VPCCConnection`. Any additional parameters after the *region_name* are passed on to the connect method of the region object.

Type *str*

Parameters *region_name* – The name of the region to connect to.

Return type `boto.vpc.VPCCConnection` or *None*

Returns A connection to the given region, or *None* if an invalid region name is given

`boto.vpc.regions` (***kw_params*)

Get all available regions for the EC2 service. You may pass any of the arguments accepted by the `VPCCConnection` object's constructor as keyword arguments and they will be passed along to the `VPCCConnection` object.

Return type *list*

Returns A list of `boto.ec2.regioninfo.RegionInfo`

boto.vpc.customergateway

Represents a Customer Gateway

class `boto.vpc.customergateway.CustomerGateway` (*connection=None*)

endElement (*name, value, connection*)

boto.vpc.dhcpoptions

Represents a DHCP Options set

class `boto.vpc.dhcpoptions.DhcpConfigSet`

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.vpc.dhcpoptions.DhcpOptions` (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.vpc.dhcpoptions.DhcpValueSet`

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.vpc.internetgateway

Represents an Internet Gateway

class `boto.vpc.internetgateway.InternetGateway` (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.vpc.internetgateway.**InternetGatewayAttachment** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.vpc.routetable

Represents a Route Table

class boto.vpc.routetable.**Route** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.vpc.routetable.**RouteAssociation** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class boto.vpc.routetable.**RouteTable** (*connection=None*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.vpc.subnet

Represents a Subnet

class boto.vpc.subnet.**Subnet** (*connection=None*)

endElement (*name, value, connection*)

boto.vpc.vpc

Represents a Virtual Private Cloud.

class boto.vpc.vpc.**VPC** (*connection=None*)

Represents a VPC.

Variables

- **id** – The unique ID of the VPC.
- **dhcp_options_id** – The ID of the set of DHCP options you've associated with the VPC (or default if the default options are associated with the VPC).
- **state** – The current state of the VPC.
- **cidr_block** – The CIDR block for the VPC.

- **is_default** – Indicates whether the VPC is the default VPC.
- **instance_tenancy** – The allowed tenancy of instances launched into the VPC.
- **classic_link_enabled** – Indicates whether ClassicLink is enabled.

attach_classic_instance (*instance_id, groups, dry_run=False*)

Links an EC2-Classic instance to a ClassicLink-enabled VPC through one or more of the VPC's security groups. You cannot link an EC2-Classic instance to more than one VPC at a time. You can only link an instance that's in the running state. An instance is automatically unlinked from a VPC when it's stopped. You can link it to the VPC again when you restart it.

After you've linked an instance, you cannot change the VPC security groups that are associated with it. To change the security groups, you must first unlink the instance, and then link it again.

Linking your instance to a VPC is sometimes referred to as attaching your instance.

Parameters

- **instance_id** – The ID of a ClassicLink-enabled VPC.
- **groups** – The ID of one or more of the VPC's security groups. You cannot specify security groups from a different VPC. The members of the list can be `boto.ec2.securitygroup.SecurityGroup` objects or strings of the id's of the security groups.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Type groups list

Return type `bool`

Returns True if successful

delete ()

detach_classic_instance (*instance_id, dry_run=False*)

Unlinks a linked EC2-Classic instance from a VPC. After the instance has been unlinked, the VPC security groups are no longer associated with it. An instance is automatically unlinked from a VPC when it's stopped.

Parameters

- **instance_id** – The ID of the VPC to which the instance is linked.
- **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

disable_classic_link (*dry_run=False*)

Disables ClassicLink for a VPC. You cannot disable ClassicLink for a VPC that has EC2-Classic instances linked to it.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

enable_classic_link (*dry_run=False*)

Enables a VPC for ClassicLink. You can then link EC2-Classic instances to your ClassicLink-enabled VPC to allow communication over private IP addresses. You cannot enable your VPC for ClassicLink if any of your VPC's route tables have existing routes for address ranges within the 10.0.0.0/8 IP address range, excluding local routes for VPCs in the 10.0.0.0/16 and 10.1.0.0/16 IP address ranges.

Parameters **dry_run** (*bool*) – Set to True if the operation should not actually run.

Return type `bool`

Returns True if successful

endElement (*name, value, connection*)

update (*validate=False, dry_run=False*)

update_classic_link_enabled (*validate=False, dry_run=False*)

Updates instance's `classic_link_enabled` attribute

Return type `bool`

Returns `self.classic_link_enabled` after update has occurred.

boto.vpc.vpnconnection

class `boto.vpc.vpnconnection.VpnConnection` (*connection=None*)

Represents a VPN Connection

Variables

- **id** – The ID of the VPN connection.
- **state** – The current state of the VPN connection. Valid values: `pending` | `available` | `deleting` | `deleted`
- **customer_gateway_configuration** – The configuration information for the VPN connection's customer gateway (in the native XML format). This element is always present in the `boto.vpc.VPCConnection.create_vpn_connection` response; however, it's present in the `boto.vpc.VPCConnection.get_all_vpn_connections` response only if the VPN connection is in the `pending` or `available` state.
- **type** – The type of VPN connection (`ipsec.1`).
- **customer_gateway_id** – The ID of the customer gateway at your end of the VPN connection.
- **vpn_gateway_id** – The ID of the virtual private gateway at the AWS side of the VPN connection.
- **tunnels** – A list of the vpn tunnels (always 2)
- **options** – The option set describing the VPN connection.
- **static_routes** – A list of static routes associated with a VPN connection.

delete (*dry_run=False*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.vpc.vpnconnection.VpnConnectionOptions` (*static_routes_only=None*)

Represents VPN connection options

Variables **static_routes_only** – Indicates whether the VPN connection uses static routes only. Static routes must be used for devices that don't support BGP.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

class `boto.vpc.vpnconnection.VpnStaticRoute` (*destination_cidr_block=None, source=None, state=None*)

Represents a static route for a VPN connection.

Variables

- **destination_cidr_block** – The CIDR block associated with the local subnet of the customer data center.
- **source** – Indicates how the routes were provided.
- **state** – The current state of the static route.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

```
class boto.vpc.vpnconnection.VpnTunnel (outside_ip_address=None, status=None,  
last_status_change=None, status_message=None,  
accepted_route_count=None)
```

Represents telemetry for a VPN tunnel

Variables

- **outside_ip_address** – The Internet-routable IP address of the virtual private gateway's outside interface.
- **status** – The status of the VPN tunnel. Valid values: UP | DOWN
- **last_status_change** – The date and time of the last change in status.
- **status_message** – If an error occurs, a description of the error.
- **accepted_route_count** – The number of accepted routes.

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

boto.vpc.vpngateway

Represents a Vpn Gateway

```
class boto.vpc.vpngateway.Attachment (connection=None)
```

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

```
class boto.vpc.vpngateway.VpnGateway (connection=None)
```

attach (*vpc_id, dry_run=False*)

endElement (*name, value, connection*)

startElement (*name, attrs, connection*)

4.139 KMS

4.139.1 boto.kms

```
boto.kms.connect_to_region (region_name, **kw_params)
```

```
boto.kms.regions ()
```

Get all available regions for the AWS Key Management Service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.139.2 boto.kms.layer1

class boto.kms.layer1.KMSConnection (**kwargs)

AWS Key Management Service (KMS) is an encryption and key management web service. This guide describes the KMS actions that you can call programmatically. For general information about KMS, see (need an address here). For the KMS developer guide, see (need address here).

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to KMS and AWS. For example, the SDKs take care of tasks such as signing requests (see below), managing errors, and retrying requests automatically. For more information about the AWS SDKs, including how to download and install them, see [‘Tools for Amazon Web Services’_](#).

We recommend that you use the AWS SDKs to make programmatic API calls to KMS. However, you can also use the KMS Query API to make direct calls to the KMS web service.

Signing Requests

Requests must be signed by using an access key ID and a secret access key. We strongly recommend that you do not use your AWS account access key ID and secret key for everyday work with KMS. Instead, use the access key ID and secret access key for an IAM user, or you can use the AWS Security Token Service to generate temporary security credentials that you can use to sign requests.

All KMS operations require [‘Signature Version 4’_](#).

Recording API Requests

KMS supports AWS CloudTrail, a service that records AWS API calls and related events for your AWS account and delivers them to an Amazon S3 bucket that you specify. By using the information collected by CloudTrail, you can determine what requests were made to KMS, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the [‘AWS CloudTrail User Guide’_](#)

Additional Resources

For more information about credentials and request signing, see the following:

- [‘AWS Security Credentials’_](#). This topic provides general information about the types of credentials used for accessing AWS.
- [‘AWS Security Token Service’_](#). This guide describes how to create and use temporary security credentials.
- [‘Signing AWS API Requests’_](#). This set of topics walks you through the process of signing a request using an access key ID and a secret access key.

```
APIVersion = '2014-11-01'
```

```
DefaultRegionEndpoint = 'kms.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
ResponseError
```

```
alias of boto.exception.JSONResponseError
```

```
ServiceName = 'KMS'
```

```
TargetPrefix = 'TrentService'
```

```
create_alias (alias_name, target_key_id)
```

Creates a display name for a customer master key. An alias can be used to identify a key and should be unique. The console enforces a one-to-one mapping between the alias and a key. An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word “alias” followed by a forward slash (alias/). An alias that begins with “aws” after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

Parameters

- **alias_name** (*string*) – String that contains the display name. Aliases that begin with AWS are reserved.
- **target_key_id** (*string*) – An identifier of the key for which you are creating the alias. This value cannot be another alias.

create_grant (*key_id, grantee_principal, retiring_principal=None, operations=None, constraints=None, grant_tokens=None*)

Adds a grant to a key to specify who can access the key and under what conditions. Grants are alternate permission mechanisms to key policies. If absent, access to the key is evaluated based on IAM policies attached to the user. By default, grants do not expire. Grants can be listed, retired, or revoked as indicated by the following APIs. Typically, when you are finished using a grant, you retire it. When you want to end a grant immediately, revoke it. For more information about grants, see **‘Grants’**.

1. ListGrants
2. RetireGrant
3. RevokeGrant

Parameters

- **key_id** (*string*) – A unique key identifier for a customer master key. This value can be a globally unique identifier, an ARN, or an alias.
- **grantee_principal** (*string*) – Principal given permission by the grant to use the key identified by the *keyId* parameter.
- **retiring_principal** (*string*) – Principal given permission to retire the grant. For more information, see RetireGrant.
- **operations** (*list*) – List of operations permitted by the grant. This can be any combination of one or more of the following values:

1. Decrypt
2. Encrypt
3. GenerateDataKey
4. GenerateDataKeyWithoutPlaintext
5. ReEncryptFrom
6. ReEncryptTo
7. CreateGrant

Parameters

- **constraints** (*dict*) – Specifies the conditions under which the actions specified by the *Operations* parameter are allowed.
- **grant_tokens** (*list*) – List of grant tokens.

create_key (*policy=None, description=None, key_usage=None*)

Creates a customer master key. Customer master keys can be used to encrypt small amounts of data (less than 4K) directly, but they are most commonly used to encrypt or envelope data keys that are then used to encrypt customer data. For more information about data keys, see GenerateDataKey and GenerateDataKeyWithoutPlaintext.

Parameters

- **policy** (*string*) – Policy to be attached to the key. This is required and delegates back to the account. The key is the root of trust.
- **description** (*string*) – Description of the key. We recommend that you choose a description that helps your customer decide whether the key is appropriate for a task.

- **key_usage** (*string*) – Specifies the intended use of the key. Currently this defaults to ENCRYPT/DECRYPT, and only symmetric encryption and decryption are supported.

decrypt (*ciphertext_blob*, *encryption_context=None*, *grant_tokens=None*)

Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted by using the Encrypt function.

Parameters

- **ciphertext_blob** (*blob*) – Ciphertext including metadata.
- **encryption_context** (*map*) – The encryption context. If this was specified in the Encrypt function, it must be specified here or the decryption operation will fail. For more information, see **‘Encryption Context’**.
- **grant_tokens** (*list*) – A list of grant tokens that represent grants which can be used to provide long term permissions to perform decryption.

delete_alias (*alias_name*)

Deletes the specified alias.

Parameters **alias_name** (*string*) – The alias to be deleted.

describe_key (*key_id*)

Provides detailed information about the specified customer master key.

Parameters **key_id** (*string*) – Unique identifier of the customer master key to be described. This can be an ARN, an alias, or a globally unique identifier.

disable_key (*key_id*)

Marks a key as disabled, thereby preventing its use.

Parameters **key_id** (*string*) – Unique identifier of the customer master key to be disabled. This can be an ARN, an alias, or a globally unique identifier.

disable_key_rotation (*key_id*)

Disables rotation of the specified key.

Parameters **key_id** (*string*) – Unique identifier of the customer master key for which rotation is to be disabled. This can be an ARN, an alias, or a globally unique identifier.

enable_key (*key_id*)

Marks a key as enabled, thereby permitting its use. You can have up to 25 enabled keys at one time.

Parameters **key_id** (*string*) – Unique identifier of the customer master key to be enabled. This can be an ARN, an alias, or a globally unique identifier.

enable_key_rotation (*key_id*)

Enables rotation of the specified customer master key.

Parameters **key_id** (*string*) – Unique identifier of the customer master key for which rotation is to be enabled. This can be an ARN, an alias, or a globally unique identifier.

encrypt (*key_id*, *plaintext*, *encryption_context=None*, *grant_tokens=None*)

Encrypts plaintext into ciphertext by using a customer master key.

Parameters

- **key_id** (*string*) – Unique identifier of the customer master. This can be an ARN, an alias, or the Key ID.
- **plaintext** (*blob*) – Data to be encrypted.
- **encryption_context** (*map*) – Name:value pair that specifies the encryption context to be used for authenticated encryption. For more information, see **‘Authenticated Encryption’**.

- **grant_tokens** (*list*) – A list of grant tokens that represent grants which can be used to provide long term permissions to perform encryption.

generate_data_key (*key_id*, *encryption_context=None*, *number_of_bytes=None*, *key_spec=None*, *grant_tokens=None*)

Generates a secure data key. Data keys are used to encrypt and decrypt data. They are wrapped by customer master keys.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.
- **encryption_context** (*map*) – Name/value pair that contains additional data to be authenticated during the encryption and decryption processes that use the key. This value is logged by AWS CloudTrail to provide context around the data encrypted by the key.
- **number_of_bytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. 1024 is the current limit.
- **key_spec** (*string*) – Value that identifies the encryption algorithm and key size to generate a data key for. Currently this can be AES_128 or AES_256.
- **grant_tokens** (*list*) – A list of grant tokens that represent grants which can be used to provide long term permissions to generate a key.

generate_data_key_without_plaintext (*key_id*, *encryption_context=None*, *key_spec=None*, *number_of_bytes=None*, *grant_tokens=None*)

Returns a key wrapped by a customer master key without the plaintext copy of that key. To retrieve the plaintext, see `GenerateDataKey`.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.
- **encryption_context** (*map*) – Name:value pair that contains additional data to be authenticated during the encryption and decryption processes.
- **key_spec** (*string*) – Value that identifies the encryption algorithm and key size. Currently this can be AES_128 or AES_256.
- **number_of_bytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on.
- **grant_tokens** (*list*) – A list of grant tokens that represent grants which can be used to provide long term permissions to generate a key.

generate_random (*number_of_bytes=None*)

Generates an unpredictable byte string.

Parameters **number_of_bytes** (*integer*) – Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. The current limit is 1024 bytes.

get_key_policy (*key_id*, *policy_name*)

Retrieves a policy attached to the specified key.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.

- **policy_name** (*string*) – String that contains the name of the policy. Currently, this must be “default”. Policy names can be discovered by calling ListKeyPolicies.

get_key_rotation_status (*key_id*)

Retrieves a Boolean value that indicates whether key rotation is enabled for the specified key.

Parameters **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.

list_aliases (*limit=None, marker=None*)

Lists all of the key aliases in the account.

Parameters

- **limit** (*integer*) – Specify this parameter when paginating results to indicate the maximum number of aliases you want in each response. If there are additional aliases beyond the maximum you specify, the *Truncated* response element will be set to *true*.
- **marker** (*string*) – Use this parameter when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the *NextMarker* element in the response you just received.

list_grants (*key_id, limit=None, marker=None*)

List the grants for a specified key.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.
- **limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of grants you want listed in the response. If there are additional grants beyond the maximum you specify, the *Truncated* response element will be set to *true*.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the *NextMarker* in the response you just received.

list_key_policies (*key_id, limit=None, marker=None*)

Retrieves a list of policies attached to a key.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.
- **limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of policies you want listed in the response. If there are additional policies beyond the maximum you specify, the *Truncated* response element will be set to *true*.
- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the *NextMarker* in the response you just received.

list_keys (*limit=None, marker=None*)

Lists the customer master keys.

Parameters

- **limit** (*integer*) – Specify this parameter only when paginating results to indicate the maximum number of keys you want listed in the response. If there

are additional keys beyond the maximum you specify, the *Truncated* response element will be set to *true*.

- **marker** (*string*) – Use this parameter only when paginating results, and only in a subsequent request after you’ve received a response where the results are truncated. Set it to the value of the *NextMarker* in the response you just received.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

put_key_policy (*key_id, policy_name, policy*)

Attaches a policy to the specified key.

Parameters

- **key_id** (*string*) – Unique identifier of the key. This can be an ARN, an alias, or a globally unique identifier.
- **policy_name** (*string*) – Name of the policy to be attached. Currently, the only supported name is “default”.
- **policy** (*string*) – The policy, in JSON format, to be attached to the key.

re_encrypt (*ciphertext_blob, destination_key_id, source_encryption_context=None, destination_encryption_context=None, grant_tokens=None*)

Encrypts data on the server side with a new customer master key without exposing the plaintext of the data on the client side. The data is first decrypted and then encrypted. This operation can also be used to change the encryption context of a ciphertext.

Parameters

- **ciphertext_blob** (*blob*) – Ciphertext of the data to re-encrypt.
- **source_encryption_context** (*map*) – Encryption context used to encrypt and decrypt the data specified in the *CiphertextBlob* parameter.
- **destination_key_id** (*string*) – Key identifier of the key used to re-encrypt the data.
- **destination_encryption_context** (*map*) – Encryption context to be used when the data is re-encrypted.
- **grant_tokens** (*list*) – Grant tokens that identify the grants that have permissions for the encryption and decryption process.

retire_grant (*grant_token*)

Retires a grant. You can retire a grant when you’re done using it to clean up. You should revoke a grant when you intend to actively deny operations that depend on it.

Parameters **grant_token** (*string*) – Token that identifies the grant to be retired.

revoke_grant (*key_id, grant_id*)

Revokes a grant. You can revoke a grant to actively deny operations that depend on it.

Parameters

- **key_id** (*string*) – Unique identifier of the key associated with the grant.
- **grant_id** (*string*) – Identifier of the grant to be revoked.

update_key_description (*key_id, description*)

Parameters

- **key_id** (*string*) –
- **description** (*string*) –

4.139.3 boto.kms.exceptions

exception boto.kms.exceptions.**AlreadyExistsException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**DependencyTimeoutException** (*status*, *reason*,
body=None, **args*)

exception boto.kms.exceptions.**DisabledException** (*status*, *reason*, *body=None*, **args*)

exception boto.kms.exceptions.**InvalidAliasNameException** (*status*, *reason*,
body=None, **args*)

exception boto.kms.exceptions.**InvalidArnException** (*status*, *reason*, *body=None*, **args*)

exception boto.kms.exceptions.**InvalidCiphertextException** (*status*, *reason*,
body=None, **args*)

exception boto.kms.exceptions.**InvalidGrantTokenException** (*status*, *reason*,
body=None, **args*)

exception boto.kms.exceptions.**InvalidKeyUsageException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**InvalidMarkerException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**KMSInternalException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**KeyUnavailableException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**LimitExceededException** (*status*, *reason*, *body=None*,
**args*)

exception boto.kms.exceptions.**MalformedPolicyDocumentException** (*status*, *reason*,
body=None,
**args*)

exception boto.kms.exceptions.**NotFoundException** (*status*, *reason*, *body=None*, **args*)

exception boto.kms.exceptions.**UnsupportedOperationException** (*status*, *reason*,
body=None, **args*)

4.140 CloudWatch Logs

4.140.1 boto.logs

boto.logs.**connect_to_region** (*region_name*, ***kw_params*)

boto.logs.**regions** ()

Get all available regions for the CloudWatch Logs service.

Return type list

Returns A list of boto.regioninfo.RegionInfo

4.140.2 boto.logs.layer1

class boto.logs.layer1.**CloudWatchLogsConnection** (***kwargs*)

Amazon CloudWatch Logs Service API Reference This is the Amazon CloudWatch Logs API Reference .
Amazon CloudWatch Logs is a managed service for real time monitoring and archival of application logs.

This guide provides detailed information about Amazon CloudWatch Logs actions, data types, parameters, and errors. For detailed information about Amazon CloudWatch Logs features and their associated API calls, go to the [‘Amazon CloudWatch Logs Developer Guide’](#).

Use the following links to get started using the Amazon CloudWatch API Reference :

- [‘Actions’](#): An alphabetical list of all Amazon CloudWatch Logs actions.
- [‘Data Types’](#): An alphabetical list of all Amazon CloudWatch Logs data types.
- [‘Common Parameters’](#): Parameters that all Query actions can use.
- [‘Common Errors’](#): Client and server errors that all actions can return.
- [‘Regions and Endpoints’](#): Itemized regions and endpoints for all AWS products.

In addition to using the Amazon CloudWatch Logs API, you can also use the following SDKs and third-party libraries to access Amazon CloudWatch Logs programmatically.

- [‘AWS SDK for Java Documentation’](#)
- [‘AWS SDK for .NET Documentation’](#)
- [‘AWS SDK for PHP Documentation’](#)
- [‘AWS SDK for Ruby Documentation’](#)

Developers in the AWS developer community also provide their own libraries, which you can find at the following AWS developer centers:

- [‘AWS Java Developer Center’](#)
- [‘AWS PHP Developer Center’](#)
- [‘AWS Python Developer Center’](#)
- [‘AWS Ruby Developer Center’](#)
- [‘AWS Windows and .NET Developer Center’](#)

```
APIVersion = '2014-03-28'
```

```
DefaultRegionEndpoint = 'logs.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
ResponseError
```

```
alias of boto.exception.JSONResponseError
```

```
ServiceName = 'CloudWatchLogs'
```

```
TargetPrefix = 'Logs_20140328'
```

```
create_log_group (log_group_name)
```

Creates a new log group with the specified name. The name of the log group must be unique within a region for an AWS account. You can create up to 100 log groups per account.

You must use the following guidelines when naming a log group:

- Log group names can be between 1 and 512 characters long.
- Allowed characters are az, AZ, 09, ‘_’ (underscore), ‘-’ (hyphen), ‘/’ (forward slash), and ‘.’ (period).

Log groups are created with a default retention of 14 days. The retention attribute allow you to configure the number of days you want to retain log events in the specified log group. See the *SetRetention* operation on how to modify the retention of your log groups.

Parameters `log_group_name` (*string*) –

```
create_log_stream (log_group_name, log_stream_name)
```

Creates a new log stream in the specified log group. The name of the log stream must be unique within the log group. There is no limit on the number of log streams that can exist in a log group.

You must use the following guidelines when naming a log stream:

- Log stream names can be between 1 and 512 characters long.
- The ‘:’ colon character is not allowed.

Parameters

- `log_group_name` (*string*) –

- **log_stream_name** (*string*) –

delete_log_group (*log_group_name*)

Deletes the log group with the specified name. Amazon CloudWatch Logs will delete a log group only if there are no log streams and no metric filters associated with the log group. If this condition is not satisfied, the request will fail and the log group will not be deleted.

Parameters **log_group_name** (*string*) –

delete_log_stream (*log_group_name*, *log_stream_name*)

Deletes a log stream and permanently deletes all the archived log events associated with it.

Parameters

- **log_group_name** (*string*) –
- **log_stream_name** (*string*) –

delete_metric_filter (*log_group_name*, *filter_name*)

Deletes a metric filter associated with the specified log group.

Parameters

- **log_group_name** (*string*) –
- **filter_name** (*string*) – The name of the metric filter.

delete_retention_policy (*log_group_name*)

Parameters **log_group_name** (*string*) –

describe_log_groups (*log_group_name_prefix=None*, *next_token=None*, *limit=None*)

Returns all the log groups that are associated with the AWS account making the request. The list returned in the response is ASCII-sorted by log group name.

By default, this operation returns up to 50 log groups. If there are more log groups to list, the response would contain a *nextToken* value in the response body. You can also limit the number of log groups returned in the response by specifying the *limit* parameter in the request.

Parameters

- **log_group_name_prefix** (*string*) –
- **next_token** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous *DescribeLogGroups* request.
- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

describe_log_streams (*log_group_name*, *log_stream_name_prefix=None*, *next_token=None*, *limit=None*)

Returns all the log streams that are associated with the specified log group. The list returned in the response is ASCII-sorted by log stream name.

By default, this operation returns up to 50 log streams. If there are more log streams to list, the response would contain a *nextToken* value in the response body. You can also limit the number of log streams returned in the response by specifying the *limit* parameter in the request.

Parameters

- **log_group_name** (*string*) –
- **log_stream_name_prefix** (*string*) –
- **next_token** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous *DescribeLogStreams* request.

- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

describe_metric_filters (*log_group_name*, *filter_name_prefix=None*, *next_token=None*, *limit=None*)

Returns all the metrics filters associated with the specified log group. The list returned in the response is ASCII-sorted by filter name.

By default, this operation returns up to 50 metric filters. If there are more metric filters to list, the response would contain a *nextToken* value in the response body. You can also limit the number of metric filters returned in the response by specifying the *limit* parameter in the request.

Parameters

- **log_group_name** (*string*) –
- **filter_name_prefix** (*string*) – The name of the metric filter.
- **next_token** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous *DescribeMetricFilters* request.
- **limit** (*integer*) – The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

get_log_events (*log_group_name*, *log_stream_name*, *start_time=None*, *end_time=None*, *next_token=None*, *limit=None*, *start_from_head=None*)

Retrieves log events from the specified log stream. You can provide an optional time range to filter the results on the event *timestamp*.

By default, this operation returns as much log events as can fit in a response size of 1MB, up to 10,000 log events. The response will always include a *nextForwardToken* and a *nextBackwardToken* in the response body. You can use any of these tokens in subsequent *GetLogEvents* requests to paginate through events in either forward or backward direction. You can also limit the number of log events returned in the response by specifying the *limit* parameter in the request.

Parameters

- **log_group_name** (*string*) –
- **log_stream_name** (*string*) –
- **start_time** (*long*) – A point in time expressed as the number milliseconds since Jan 1, 1970 00:00:00 UTC.
- **end_time** (*long*) – A point in time expressed as the number milliseconds since Jan 1, 1970 00:00:00 UTC.
- **next_token** (*string*) – A string token used for pagination that points to the next page of results. It must be a value obtained from the *nextForwardToken* or *nextBackwardToken* fields in the response of the previous *GetLogEvents* request.
- **limit** (*integer*) – The maximum number of log events returned in the response. If you don't specify a value, the request would return as much log events as can fit in a response size of 1MB, up to 10,000 log events.
- **start_from_head** (*boolean*) –

make_request (*action*, *body*)

Makes a request to the server, with stock multiple-retry logic.

put_log_events (*log_group_name*, *log_stream_name*, *log_events*, *sequence_token=None*)

Uploads a batch of log events to the specified log stream.

Every `PutLogEvents` request must include the `sequenceToken` obtained from the response of the previous request. An upload in a newly created log stream does not require a `sequenceToken`.

The batch of events must satisfy the following constraints:

- The maximum batch size is 32,768 bytes, and this size is calculated as the sum of all event messages in UTF-8, plus 26 bytes for each log event.
- None of the log events in the batch can be more than 2 hours in the future.
- None of the log events in the batch can be older than 14 days or the retention period of the log group.
- The log events in the batch must be in chronological order by their `timestamp`.
- The maximum number of log events in a batch is 1,000.

Parameters

- `log_group_name` (*string*) –
- `log_stream_name` (*string*) –
- `log_events` (*list*) – A list of events belonging to a log stream.
- `sequence_token` (*string*) – A string token that must be obtained from the response of the previous `PutLogEvents` request.

put_metric_filter (*log_group_name, filter_name, filter_pattern, metric_transformations*)

Creates or updates a metric filter and associates it with the specified log group. Metric filters allow you to configure rules to extract metric data from log events ingested through `PutLogEvents` requests.

Parameters

- `log_group_name` (*string*) –
- `filter_name` (*string*) – The name of the metric filter.
- `filter_pattern` (*string*) –
- `metric_transformations` (*list*) –

put_retention_policy (*log_group_name, retention_in_days*)

Parameters

- `log_group_name` (*string*) –
- `retention_in_days` (*integer*) – Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 547, 730.

set_retention (*log_group_name, retention_in_days*)

Sets the retention of the specified log group. Log groups are created with a default retention of 14 days. The retention attribute allow you to configure the number of days you want to retain log events in the specified log group.

Parameters

- `log_group_name` (*string*) –
- `retention_in_days` (*integer*) – Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 547, 730.

test_metric_filter (*filter_pattern, log_event_messages*)

Tests the filter pattern of a metric filter against a sample of log event messages. You can use this operation to validate the correctness of a metric filter pattern.

Parameters

- `filter_pattern` (*string*) –

- `log_event_messages` (*list*) –

4.140.3 boto.logs.exceptions

exception `boto.logs.exceptions.DataAlreadyAcceptedException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.InvalidParameterException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.InvalidSequenceTokenException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.LimitExceededException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.OperationAbortedException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.ResourceAlreadyExistsException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.ResourceInUseException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.ResourceNotFoundException` (*status*, *reason*, *body=None*, **args*)

exception `boto.logs.exceptions.ServiceUnavailableException` (*status*, *reason*, *body=None*, **args*)

4.141 Machine Learning

4.141.1 boto.machinelearning

`boto.machinelearning.connect_to_region` (*region_name*, ***kw_params*)

`boto.machinelearning.regions` ()

Get all available regions for the Amazon Machine Learning.

Return type *list*

Returns A list of `boto.regioninfo.RegionInfo`

4.141.2 boto.machinelearning.layer1

class `boto.machinelearning.layer1.MachineLearningConnection` (***kwargs*)

Definition of the public APIs exposed by Amazon Machine Learning

APIVersion = '2014-12-12'

AuthServiceName = 'machinelearning'

DefaultRegionEndpoint = 'machinelearning.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

```
ServiceName = 'MachineLearning'
```

```
TargetPrefix = 'AmazonML_20141212'
```

```
create_batch_prediction(batch_prediction_id, ml_model_id,  
                        batch_prediction_data_source_id, output_uri,  
                        batch_prediction_name=None)
```

Generates predictions for a group of observations. The observations to process exist in one or more data files referenced by a *DataSource*. This operation creates a new *BatchPrediction*, and uses an *MLModel* and the data files referenced by the *DataSource* as information sources.

CreateBatchPrediction is an asynchronous operation. In response to *CreateBatchPrediction*, Amazon Machine Learning (Amazon ML) immediately returns and sets the *BatchPrediction* status to *PENDING*. After the *BatchPrediction* completes, Amazon ML sets the status to *COMPLETED*.

You can poll for status updates by using the *GetBatchPrediction* operation and checking the *Status* parameter of the result. After the *COMPLETED* status appears, the results are available in the location specified by the *OutputUri* parameter.

Parameters

- **batch_prediction_id** (*string*) – A user-supplied ID that uniquely identifies the *BatchPrediction*.
- **batch_prediction_name** (*string*) – A user-supplied name or description of the *BatchPrediction*. *BatchPredictionName* can only use the UTF-8 character set.
- **ml_model_id** (*string*) – The ID of the *MLModel* that will generate predictions for the group of observations.
- **batch_prediction_data_source_id** (*string*) – The ID of the *DataSource* that points to the group of observations to predict.
- **output_uri** (*string*) – The location of an Amazon Simple Storage Service (Amazon S3) bucket or directory to store the batch prediction results. The following substrings are not allowed in the s3 key portion of the “outputURI” field: ‘:’, ‘/’, ‘.’, ‘..’.

Amazon ML needs permissions to store and retrieve the logs on your behalf. For information about how to set permissions, see the [‘Amazon Machine Learning Developer Guide’](#).

```
create_data_source_from_rds(data_source_id, rds_data, role_arn, data_source_name=None,  
                            compute_statistics=None)
```

Creates a *DataSource* object from an ‘Amazon Relational Database Service’ (Amazon RDS). A *DataSource* references data that can be used to perform *CreateMLModel*, *CreateEvaluation*, or *CreateBatchPrediction* operations.

CreateDataSourceFromRDS is an asynchronous operation. In response to *CreateDataSourceFromRDS*, Amazon Machine Learning (Amazon ML) immediately returns and sets the *DataSource* status to *PENDING*. After the *DataSource* is created and ready for use, Amazon ML sets the *Status* parameter to *COMPLETED*. *DataSource* in *COMPLETED* or *PENDING* status can only be used to perform *CreateMLModel*, *CreateEvaluation*, or *CreateBatchPrediction* operations.

If Amazon ML cannot accept the input source, it sets the *Status* parameter to *FAILED* and includes an error message in the *Message* attribute of the *GetDataSource* operation response.

Parameters

- **data_source_id** (*string*) – A user-supplied ID that uniquely identifies the *DataSource*. Typically, an Amazon Resource Number (ARN) becomes the ID for a *DataSource*.

- **data_source_name** (*string*) – A user-supplied name or description of the *DataSource*.
- **rds_data** (*dict*) –

The data specification of an Amazon RDS *DataSource*:

- DatabaseInformation -
 - ‘DatabaseName’ - Name of the Amazon RDS database.
 - ‘InstanceIdentifier’ - Unique identifier for the Amazon RDS database instance.
- DatabaseCredentials - AWS Identity and Access Management (IAM) credentials that are used to connect to the Amazon RDS database.
- ResourceRole - Role (DataPipelineDefaultResourceRole) assumed by an Amazon Elastic Compute Cloud (EC2) instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see ‘Role templates’_ for data pipelines.
- ServiceRole - Role (DataPipelineDefaultRole) assumed by the AWS Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon Simple Storage Service (S3). For more information, see ‘Role templates’_ for data pipelines.
- SecurityInfo - Security information to use to access an Amazon RDS instance. You need to set up appropriate ingress rules for the security entity IDs provided to allow access to the Amazon RDS instance. Specify a [SubnetId, SecurityGroupIds] pair for a VPC-based Amazon RDS instance.
- SelectSqlQuery - Query that is used to retrieve the observation data for the *DataSource*.
- S3StagingLocation - Amazon S3 location for staging RDS data. The data retrieved from Amazon RDS using *SelectSqlQuery* is stored in this location.
- DataSchemaUri - Amazon S3 location of the *DataSchema*.
- DataSchema - A JSON string representing the schema. This is not required if *DataSchemaUri* is specified.
- DataRearrangement - A JSON string representing the splitting requirement of a *DataSource*. Sample - ‘ “{“randomSeed”:”some-random-seed”, “splitting”:{“percentBegin”:10,”percentEnd”:60}}’“

Parameters

- **role_arn** (*string*) – The role that Amazon ML assumes on behalf of the user to create and activate a data pipeline in the users account and copy data (using the *SelectSqlQuery*) query from Amazon RDS to Amazon S3.
- **compute_statistics** (*boolean*) – The compute statistics for a *DataSource*. The statistics are generated from the observation data referenced by a *DataSource*. Amazon ML uses the statistics internally during an *MLModel* training. This parameter must be set to *True* if the ‘DataSource’ needs to be used for *MLModel* training.

create_data_source_from_redshift (*data_source_id*, *data_spec*, *role_arn*, *data_source_name=None*, *compute_statistics=None*)

Creates a *DataSource* from ‘Amazon Redshift’_. A *DataSource* references data that can be used to perform either *CreateMLModel*, *CreateEvaluation* or *CreateBatchPrediction* operations.

CreateDataSourceFromRedshift is an asynchronous operation. In response to *CreateDataSourceFromRedshift*, Amazon Machine Learning (Amazon ML) immediately returns and sets the *DataSource* status to *PENDING*. After the *DataSource* is created and ready for use, Amazon ML sets the *Status* parameter to *COMPLETED*. *DataSource* in *COMPLETED* or *PENDING* status can only be used to perform *CreateMLModel*, *CreateEvaluation*, or *CreateBatchPrediction* operations.

If Amazon ML cannot accept the input source, it sets the *Status* parameter to *FAILED* and includes an error message in the *Message* attribute of the *GetDataSource* operation response.

The observations should exist in the database hosted on an Amazon Redshift cluster and should be specified by a *SelectSqlQuery*. Amazon ML executes ‘Unload’_ command in Amazon Redshift to transfer the result set of *SelectSqlQuery* to *S3StagingLocation*.

After the *DataSource* is created, it’s ready for use in evaluations and batch predictions. If you plan to use the *DataSource* to train an *MLModel*, the *DataSource* requires another item – a recipe. A recipe describes the observation variables that participate in training an *MLModel*. A recipe describes how each input variable will be used in training. Will the variable be included or excluded from training? Will the variable be manipulated, for example, combined with another variable or split apart into word combinations? The recipe provides answers to these questions. For more information, see the Amazon Machine Learning Developer Guide.

Parameters

- **data_source_id** (*string*) – A user-supplied ID that uniquely identifies the *DataSource*.
- **data_source_name** (*string*) – A user-supplied name or description of the *DataSource*.
- **data_spec** (*dict*) –

The data specification of an Amazon Redshift *DataSource*:

- DatabaseInformation -
 - ‘DatabaseName’ - Name of the Amazon Redshift database.
 - ‘ClusterIdentifier’ - Unique ID for the Amazon Redshift cluster.
- **DatabaseCredentials - AWS Identity and Access Management (IAM)** credentials that are used to connect to the Amazon Redshift database.
- **SelectSqlQuery - Query that is used to retrieve the observation data** for the *DataSource*.
- **S3StagingLocation - Amazon Simple Storage Service (Amazon S3)** location for staging Amazon Redshift data. The data retrieved from Amazon Relational Database Service (Amazon RDS) using *SelectSqlQuery* is stored in this location.
- **DataSchemaUri** - Amazon S3 location of the *DataSchema*.
- **DataSchema - A JSON string representing the schema. This is not** required if *DataSchemaUri* is specified.
- **DataRearrangement - A JSON string representing the splitting** requirement of a *DataSource*. Sample - ‘ “{“randomSeed”:“some-random-seed”, “splitting”:{“percentBegin”:10,”percentEnd”:60}}” ‘

Parameters role_arn (*string*) – A fully specified role Amazon Resource Name (ARN). Amazon ML assumes the role on behalf of the user to create the following:

- **A security group to allow Amazon ML to execute the *SelectSqlQuery*** query on an Amazon Redshift cluster
- **An Amazon S3 bucket policy to grant Amazon ML read/write permissions** on the *S3StagingLocation*

Parameters compute_statistics (*boolean*) – The compute statistics for a *DataSource*. The statistics are generated from the observation data referenced by a *DataSource*. Amazon ML uses the statistics internally during *MLModel* training. This parameter must be set to *True* if the ‘‘DataSource’‘ needs to be used for *MLModel* training

create_data_source_from_s3 (*data_source_id*, *data_spec*, *data_source_name=None*, *compute_statistics=None*)

Creates a *DataSource* object. A *DataSource* references data that can be used to perform *CreateMLModel*, *CreateEvaluation*, or *CreateBatchPrediction* operations.

CreateDataSourceFromS3 is an asynchronous operation. In response to *CreateDataSourceFromS3*, Amazon Machine Learning (Amazon ML) immediately returns and sets the *DataSource* status to *PENDING*.

ING. After the *DataSource* is created and ready for use, Amazon ML sets the *Status* parameter to *COMPLETED*. *DataSource* in *COMPLETED* or *PENDING* status can only be used to perform *CreateMLModel*, *CreateEvaluation* or *CreateBatchPrediction* operations.

If Amazon ML cannot accept the input source, it sets the *Status* parameter to *FAILED* and includes an error message in the *Message* attribute of the *GetDataSource* operation response.

The observation data used in a *DataSource* should be ready to use; that is, it should have a consistent structure, and missing data values should be kept to a minimum. The observation data must reside in one or more CSV files in an Amazon Simple Storage Service (Amazon S3) bucket, along with a schema that describes the data items by name and type. The same schema must be used for all of the data files referenced by the *DataSource*.

After the *DataSource* has been created, it's ready to use in evaluations and batch predictions. If you plan to use the *DataSource* to train an *MLModel*, the *DataSource* requires another item: a recipe. A recipe describes the observation variables that participate in training an *MLModel*. A recipe describes how each input variable will be used in training. Will the variable be included or excluded from training? Will the variable be manipulated, for example, combined with another variable, or split apart into word combinations? The recipe provides answers to these questions. For more information, see the ‘[Amazon Machine Learning Developer Guide](#)’.

Parameters

- **data_source_id** (*string*) – A user-supplied identifier that uniquely identifies the *DataSource*.
- **data_source_name** (*string*) – A user-supplied name or description of the *DataSource*.
- **data_spec** (*dict*) –

The data specification of a *DataSource*:

- **DataLocationS3 - Amazon Simple Storage Service (Amazon S3) location** of the observation data.
- **DataSchemaLocationS3** - Amazon S3 location of the *DataSchema*.
- **DataSchema** - A JSON string representing the schema. This is not required if *DataSchemaUri* is specified.
- **DataRearrangement** - A JSON string representing the splitting requirement of a *Datasource*. Sample - ‘ “{“randomSeed”:”some-random-seed”, “splitting”:{“percentBegin”:10,”percentEnd”:60}}’ “

Parameters compute_statistics (*boolean*) – The compute statistics for a *DataSource*. The statistics are generated from the observation data referenced by a *DataSource*. Amazon ML uses the statistics internally during an *MLModel* training. This parameter must be set to *True* if the “*DataSource* “ needs to be used for *MLModel* training

create_evaluation (*evaluation_id*, *ml_model_id*, *evaluation_data_source_id*, *evaluation_name=None*)

Creates a new *Evaluation* of an *MLModel*. An *MLModel* is evaluated on a set of observations associated to a *DataSource*. Like a *DataSource* for an *MLModel*, the *DataSource* for an *Evaluation* contains values for the Target Variable. The *Evaluation* compares the predicted result for each observation to the actual outcome and provides a summary so that you know how effective the *MLModel* functions on the test data. *Evaluation* generates a relevant performance metric such as BinaryAUC, RegressionRMSE or MulticlassAvgFScore based on the corresponding *MLModelType*: *BINARY*, *REGRESSION* or *MULTICLASS*.

CreateEvaluation is an asynchronous operation. In response to *CreateEvaluation*, Amazon Machine Learning (Amazon ML) immediately returns and sets the evaluation status to *PENDING*. After the *Evaluation* is created and ready for use, Amazon ML sets the status to *COMPLETED*.

You can use the *GetEvaluation* operation to check progress of the evaluation during the creation operation.

Parameters

- **evaluation_id** (*string*) – A user-supplied ID that uniquely identifies the *Evaluation*.
- **evaluation_name** (*string*) – A user-supplied name or description of the *Evaluation*.
- **ml_model_id** (*string*) – The ID of the *MLModel* to evaluate.

The schema used in creating the *MLModel* must match the schema of the *DataSource* used in the *Evaluation*.

Parameters **evaluation_data_source_id** (*string*) – The ID of the *DataSource* for the evaluation. The schema of the *DataSource* must match the schema used to create the *MLModel*.

create_ml_model (*ml_model_id*, *ml_model_type*, *training_data_source_id*, *ml_model_name=None*, *parameters=None*, *recipe=None*, *recipe_uri=None*)
Creates a new *MLModel* using the data files and the recipe as information sources.

An *MLModel* is nearly immutable. Users can only update the *MLModelName* and the *ScoreThreshold* in an *MLModel* without creating a new *MLModel*.

CreateMLModel is an asynchronous operation. In response to *CreateMLModel*, Amazon Machine Learning (Amazon ML) immediately returns and sets the *MLModel* status to *PENDING*. After the *MLModel* is created and ready for use, Amazon ML sets the status to *COMPLETED*.

You can use the *GetMLModel* operation to check progress of the *MLModel* during the creation operation.

CreateMLModel requires a *DataSource* with computed statistics, which can be created by setting *ComputeStatistics* to *True* in *CreateDataSourceFromRDS*, *CreateDataSourceFromS3*, or *CreateDataSourceFromRedshift* operations.

Parameters

- **ml_model_id** (*string*) – A user-supplied ID that uniquely identifies the *MLModel*.
- **ml_model_name** (*string*) – A user-supplied name or description of the *MLModel*.
- **ml_model_type** (*string*) – The category of supervised learning that this *MLModel* will address. Choose from the following types:
 - Choose *REGRESSION* if the *MLModel* will be used to predict a numeric value.
 - Choose *BINARY* if the *MLModel* result has two possible values.
 - Choose *MULTICLASS* if the *MLModel* result has a limited number of values.

For more information, see the ‘Amazon Machine Learning Developer Guide’.

Parameters **parameters** (*map*) –

A list of the training parameters in the *MLModel*. The list is implemented as a map of key/value pairs.

The following is the current set of training parameters:

- **sgd.L1RegularizationAmount - Coefficient regularization L1 norm.** It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in sparse feature set. If you use this parameter, start by specifying a small value such as 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L1 normalization. The parameter cannot be used when *L2* is specified. Use this parameter sparingly.

- ***sgd.l2RegularizationAmount* - Coefficient regularization L2 norm. It** controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, start by specifying a small value such as 1.0E-08. The value is a double that ranges from 0 to MAX_DOUBLE. The default is not to use L2 normalization. This cannot be used when *L1* is specified. Use this parameter sparingly.
- ***sgd.maxPasses* - Number of times that the training process traverses** the observations to build the *MLModel*. The value is an integer that ranges from 1 to 10000. The default value is 10.
- ***sgd.maxMLModelSizeInBytes* - Maximum allowed size of the model.** Depending on the input data, the size of the model might affect its performance. The value is an integer that ranges from 100000 to 2147483648. The default value is 33554432.

Parameters

- ***training_data_source_id* (*string*)** – The *DataSource* that points to the training data.
- ***recipe* (*string*)** – The data recipe for creating *MLModel*. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.
- ***recipe_uri* (*string*)** – The Amazon Simple Storage Service (Amazon S3) location and file name that contains the *MLModel* recipe. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.

***create_realtime_endpoint* (*ml_model_id*)**

Creates a real-time endpoint for the *MLModel*. The endpoint contains the URI of the *MLModel*; that is, the location to send real-time prediction requests for the specified *MLModel*.

Parameters *ml_model_id* (*string*) – The ID assigned to the *MLModel* during creation.

***delete_batch_prediction* (*batch_prediction_id*)**

Assigns the DELETED status to a *BatchPrediction*, rendering it unusable.

After using the *DeleteBatchPrediction* operation, you can use the *GetBatchPrediction* operation to verify that the status of the *BatchPrediction* changed to DELETED.

The result of the *DeleteBatchPrediction* operation is irreversible.

Parameters *batch_prediction_id* (*string*) – A user-supplied ID that uniquely identifies the *BatchPrediction*.

***delete_data_source* (*data_source_id*)**

Assigns the DELETED status to a *DataSource*, rendering it unusable.

After using the *DeleteDataSource* operation, you can use the *GetDataSource* operation to verify that the status of the *DataSource* changed to DELETED.

The results of the *DeleteDataSource* operation are irreversible.

Parameters *data_source_id* (*string*) – A user-supplied ID that uniquely identifies the *DataSource*.

***delete_evaluation* (*evaluation_id*)**

Assigns the DELETED status to an *Evaluation*, rendering it unusable.

After invoking the *DeleteEvaluation* operation, you can use the *GetEvaluation* operation to verify that the status of the *Evaluation* changed to DELETED.

The results of the *DeleteEvaluation* operation are irreversible.

Parameters *evaluation_id* (*string*) – A user-supplied ID that uniquely identifies the *Evaluation* to delete.

delete_ml_model (*ml_model_id*)

Assigns the DELETED status to an *MLModel*, rendering it unusable.

After using the *DeleteMLModel* operation, you can use the *GetMLModel* operation to verify that the status of the *MLModel* changed to DELETED.

The result of the *DeleteMLModel* operation is irreversible.

Parameters *ml_model_id* (*string*) – A user-supplied ID that uniquely identifies the *MLModel*.

delete_realtime_endpoint (*ml_model_id*)

Deletes a real time endpoint of an *MLModel*.

Parameters *ml_model_id* (*string*) – The ID assigned to the *MLModel* during creation.

describe_batch_predictions (*filter_variable=None, eq=None, gt=None, lt=None, ge=None, le=None, ne=None, prefix=None, sort_order=None, next_token=None, limit=None*)

Returns a list of *BatchPrediction* operations that match the search criteria in the request.

Parameters *filter_variable* (*string*) –

Use one of the following variables to filter a list of *BatchPrediction*:

- **CreatedAt** - Sets the search criteria to the *BatchPrediction* creation date.
- **Status** - Sets the search criteria to the *BatchPrediction* status.
- **Name** - Sets the search criteria to the contents of the *BatchPrediction* *** *Name*.
- **IAMUser** - Sets the search criteria to the user account that invoked the *BatchPrediction* creation.
- **MLModelId** - Sets the search criteria to the *MLModel* used in the *BatchPrediction*.
- **DataSourceId** - Sets the search criteria to the *DataSource* used in the *BatchPrediction*.
- **DataURI** - Sets the search criteria to the data file(s) used in the *BatchPrediction*. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.

Parameters

- **eq** (*string*) – The equal to operator. The *BatchPrediction* results will have *FilterVariable* values that exactly match the value specified with *EQ*.
- **gt** (*string*) – The greater than operator. The *BatchPrediction* results will have *FilterVariable* values that are greater than the value specified with *GT*.
- **lt** (*string*) – The less than operator. The *BatchPrediction* results will have *FilterVariable* values that are less than the value specified with *LT*.
- **ge** (*string*) – The greater than or equal to operator. The *BatchPrediction* results will have *FilterVariable* values that are greater than or equal to the value specified with *GE*.
- **le** (*string*) – The less than or equal to operator. The *BatchPrediction* results will have *FilterVariable* values that are less than or equal to the value specified with *LE*.
- **ne** (*string*) – The not equal to operator. The *BatchPrediction* results will have *FilterVariable* values not equal to the value specified with *NE*.
- **prefix** (*string*) –

A string that is found at the beginning of a variable, such as *Name* or *Id*.

For example, a *Batch Prediction* operation could have the *Name* *2014-09-09-HolidayGiftMailer*. To search for this *BatchPrediction*, select *Name* for the *FilterVariable* and any of the following strings for the *Prefix*:

- 2014-09

- 2014-09-09
- 2014-09-09-Holiday

Parameters `sort_order` (*string*) – A two-value parameter that determines the sequence of the resulting list of `MLModel`'s.

- *asc* - Arranges the list in ascending order (A-Z, 0-9).
- *dsc* - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by *FilterVariable*.

Parameters

- `next_token` (*string*) – An ID of the page in the paginated results.
- `limit` (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

describe_data_sources (*filter_variable=None, eq=None, gt=None, lt=None, ge=None, le=None, ne=None, prefix=None, sort_order=None, next_token=None, limit=None*)

Returns a list of *DataSource* that match the search criteria in the request.

Parameters `filter_variable` (*string*) –

Use one of the following variables to filter a list of *DataSource*:

- **CreatedAt** - Sets the search criteria to *DataSource* creation dates.
- **Status** - Sets the search criteria to *DataSource* statuses.
- **Name** - Sets the search criteria to the contents of *DataSource* `** ** Name`.
- **DataUri** - Sets the search criteria to the URI of data files used to create the *DataSource*. The URI can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- **IAMUser** - Sets the search criteria to the user account that invoked the *DataSource* creation.

Parameters

- `eq` (*string*) – The equal to operator. The *DataSource* results will have *FilterVariable* values that exactly match the value specified with *EQ*.
- `gt` (*string*) – The greater than operator. The *DataSource* results will have *FilterVariable* values that are greater than the value specified with *GT*.
- `lt` (*string*) – The less than operator. The *DataSource* results will have *FilterVariable* values that are less than the value specified with *LT*.
- `ge` (*string*) – The greater than or equal to operator. The *DataSource* results will have *FilterVariable* values that are greater than or equal to the value specified with *GE*.
- `le` (*string*) – The less than or equal to operator. The *DataSource* results will have *FilterVariable* values that are less than or equal to the value specified with *LE*.
- `ne` (*string*) – The not equal to operator. The *DataSource* results will have *FilterVariable* values not equal to the value specified with *NE*.
- `prefix` (*string*) –

A string that is found at the beginning of a variable, such as *Name* or *Id*.

For example, a *DataSource* could have the *Name* `2014-09-09-HolidayGiftMailer`. To search for this *DataSource*, select *Name* for the *FilterVariable* and any of the following strings for the *Prefix*:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Parameters **sort_order** (*string*) – A two-value parameter that determines the sequence of the resulting list of *DataSource*.

- *asc* - Arranges the list in ascending order (A-Z, 0-9).
- *dsc* - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by *FilterVariable*.

Parameters

- **next_token** (*string*) – The ID of the page in the paginated results.
- **limit** (*integer*) – The maximum number of *DataSource* to include in the result.

describe_evaluations (*filter_variable=None, eq=None, gt=None, lt=None, ge=None, le=None, ne=None, prefix=None, sort_order=None, next_token=None, limit=None*)

Returns a list of *DescribeEvaluations* that match the search criteria in the request.

Parameters **filter_variable** (*string*) –

Use one of the following variable to filter a list of *Evaluation* objects:

- **CreatedAt** - Sets the search criteria to the *Evaluation* creation date.
- **Status** - Sets the search criteria to the *Evaluation* status.
- **Name** - Sets the search criteria to the contents of *Evaluation* ** *Name.
- **IAMUser** - Sets the search criteria to the user account that invoked an *Evaluation*.
- **MLModelId** - Sets the search criteria to the *MLModel* that was evaluated.
- **DataSourceId** - Sets the search criteria to the *DataSource* used in *Evaluation*.
- **DataUri** - Sets the search criteria to the data file(s) used in *Evaluation*. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.

Parameters

- **eq** (*string*) – The equal to operator. The *Evaluation* results will have *FilterVariable* values that exactly match the value specified with *EQ*.
- **gt** (*string*) – The greater than operator. The *Evaluation* results will have *FilterVariable* values that are greater than the value specified with *GT*.
- **lt** (*string*) – The less than operator. The *Evaluation* results will have *FilterVariable* values that are less than the value specified with *LT*.
- **ge** (*string*) – The greater than or equal to operator. The *Evaluation* results will have *FilterVariable* values that are greater than or equal to the value specified with *GE*.
- **le** (*string*) – The less than or equal to operator. The *Evaluation* results will have *FilterVariable* values that are less than or equal to the value specified with *LE*.
- **ne** (*string*) – The not equal to operator. The *Evaluation* results will have *FilterVariable* values not equal to the value specified with *NE*.
- **prefix** (*string*) –

A string that is found at the beginning of a variable, such as *Name* or *Id*.

For example, an *Evaluation* could have the *Name* *2014-09-09-HolidayGiftMailer*. To search for this *Evaluation*, select *Name* for the *FilterVariable* and any of the following strings for the *Prefix*:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Parameters `sort_order` (*string*) – A two-value parameter that determines the sequence of the resulting list of *Evaluation*.

- *asc* - Arranges the list in ascending order (A-Z, 0-9).
- *dsc* - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by *FilterVariable*.

Parameters

- `next_token` (*string*) – The ID of the page in the paginated results.
- `limit` (*integer*) – The maximum number of *Evaluation* to include in the result.

describe_ml_models (*filter_variable=None, eq=None, gt=None, lt=None, ge=None, le=None, ne=None, prefix=None, sort_order=None, next_token=None, limit=None*)
Returns a list of *MLModel* that match the search criteria in the request.

Parameters `filter_variable` (*string*) –

Use one of the following variables to filter a list of *MLModel*:

- *CreatedAt* - Sets the search criteria to *MLModel* creation date.
- *Status* - Sets the search criteria to *MLModel* status.
- **Name** - Sets the search criteria to the contents of *MLModel* ** * *Name*.
- **IAMUser** - Sets the search criteria to the user account that invoked the *MLModel* creation.
- **TrainingDataSourceId** - Sets the search criteria to the *DataSource* used to train one or more *MLModel*.
- **RealtimeEndpointStatus** - Sets the search criteria to the *MLModel* real-time endpoint status.
- **MLModelType** - Sets the search criteria to *MLModel* type: binary, regression, or multi-class.
- **Algorithm** - Sets the search criteria to the algorithm that the *MLModel* uses.
- **TrainingDataURI** - Sets the search criteria to the data file(s) used in training a *MLModel*. The URL can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.

Parameters

- `eq` (*string*) – The equal to operator. The *MLModel* results will have *FilterVariable* values that exactly match the value specified with *EQ*.
- `gt` (*string*) – The greater than operator. The *MLModel* results will have *FilterVariable* values that are greater than the value specified with *GT*.
- `lt` (*string*) – The less than operator. The *MLModel* results will have *FilterVariable* values that are less than the value specified with *LT*.
- `ge` (*string*) – The greater than or equal to operator. The *MLModel* results will have *FilterVariable* values that are greater than or equal to the value specified with *GE*.
- `le` (*string*) – The less than or equal to operator. The *MLModel* results will have *FilterVariable* values that are less than or equal to the value specified with *LE*.
- `ne` (*string*) – The not equal to operator. The *MLModel* results will have *FilterVariable* values not equal to the value specified with *NE*.
- `prefix` (*string*) –

A string that is found at the beginning of a variable, such as *Name* or *Id*.

For example, an *MLModel* could have the *Name* *2014-09-09-HolidayGiftMailer*. To search for this *MLModel*, select *Name* for the *FilterVariable* and any of the following strings for the *Prefix*:

- 2014-09
- 2014-09-09

- 2014-09-09-Holiday

Parameters `sort_order` (*string*) – A two-value parameter that determines the sequence of the resulting list of *MLModel*.

- *asc* - Arranges the list in ascending order (A-Z, 0-9).
- *dsc* - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by *FilterVariable*.

Parameters

- `next_token` (*string*) – The ID of the page in the paginated results.
- `limit` (*integer*) – The number of pages of information to include in the result. The range of acceptable values is 1 through 100. The default value is 100.

get_batch_prediction (*batch_prediction_id*)

Returns a *BatchPrediction* that includes detailed metadata, status, and data file information for a *BatchPrediction* request.

Parameters `batch_prediction_id` (*string*) – An ID assigned to the *BatchPrediction* at creation.

get_data_source (*data_source_id*, *verbose=None*)

Returns a *DataSource* that includes metadata and data file information, as well as the current status of the *DataSource*.

GetDataSource provides results in normal or verbose format. The verbose format adds the schema description and the list of files pointed to by the *DataSource* to the normal format.

Parameters

- `data_source_id` (*string*) – The ID assigned to the *DataSource* at creation.
- `verbose` (*boolean*) – Specifies whether the *GetDataSource* operation should return *DataSourceSchema*.

If true, *DataSourceSchema* is returned.

If false, *DataSourceSchema* is not returned.

get_evaluation (*evaluation_id*)

Returns an *Evaluation* that includes metadata as well as the current status of the *Evaluation*.

Parameters `evaluation_id` (*string*) – The ID of the *Evaluation* to retrieve. The evaluation of each *MLModel* is recorded and cataloged. The ID provides the means to access the information.

get_ml_model (*ml_model_id*, *verbose=None*)

Returns an *MLModel* that includes detailed metadata, and data source information as well as the current status of the *MLModel*.

GetMLModel provides results in normal or verbose format.

Parameters

- `ml_model_id` (*string*) – The ID assigned to the *MLModel* at creation.
- `verbose` (*boolean*) – Specifies whether the *GetMLModel* operation should return *Recipe*.

If true, *Recipe* is returned.

If false, *Recipe* is not returned.

make_request (*action*, *body*, *host=None*)

Makes a request to the server, with stock multiple-retry logic.

predict (*ml_model_id*, *record*, *predict_endpoint*)

Generates a prediction for the observation using the specified *MLModel*.

Not all response parameters will be populated because this is dependent on the type of requested model.

Parameters

- **ml_model_id** (*string*) – A unique identifier of the *MLModel*.
- **record** (*map*) – A map of variable name-value pairs that represent an observation.
- **predict_endpoint** (*string*) – The endpoint to send the predict request to.

update_batch_prediction (*batch_prediction_id*, *batch_prediction_name*)

Updates the *BatchPredictionName* of a *BatchPrediction*.

You can use the *GetBatchPrediction* operation to view the contents of the updated data element.

Parameters

- **batch_prediction_id** (*string*) – The ID assigned to the *BatchPrediction* during creation.
- **batch_prediction_name** (*string*) – A new user-supplied name or description of the *BatchPrediction*.

update_data_source (*data_source_id*, *data_source_name*)

Updates the *DataSourceName* of a *DataSource*.

You can use the *GetDataSource* operation to view the contents of the updated data element.

Parameters

- **data_source_id** (*string*) – The ID assigned to the *DataSource* during creation.
- **data_source_name** (*string*) – A new user-supplied name or description of the *DataSource* that will replace the current description.

update_evaluation (*evaluation_id*, *evaluation_name*)

Updates the *EvaluationName* of an *Evaluation*.

You can use the *GetEvaluation* operation to view the contents of the updated data element.

Parameters

- **evaluation_id** (*string*) – The ID assigned to the *Evaluation* during creation.
- **evaluation_name** (*string*) – A new user-supplied name or description of the *Evaluation* that will replace the current content.

update_ml_model (*ml_model_id*, *ml_model_name=None*, *score_threshold=None*)

Updates the *MLModelName* and the *ScoreThreshold* of an *MLModel*.

You can use the *GetMLModel* operation to view the contents of the updated data element.

Parameters

- **ml_model_id** (*string*) – The ID assigned to the *MLModel* during creation.
- **ml_model_name** (*string*) – A user-supplied name or description of the *MLModel*.
- **score_threshold** (*float*) – The *ScoreThreshold* used in binary classification *MLModel* that marks the boundary between a positive prediction and a negative prediction.

Output values greater than or equal to the *ScoreThreshold* receive a positive result from the *MLModel*, such as *True*. Output values less than the *ScoreThreshold* receive a negative response from the *MLModel*, such as *False*.

4.141.3 boto.machinelearning.exceptions

```
exception boto.machinelearning.exceptions.IdempotentParameterMismatchException (status,
                                                                                rea-
                                                                                son,
                                                                                body=None,
                                                                                *args)
```

```
exception boto.machinelearning.exceptions.InternalServerErrorException (status,
                                                                           reason,
                                                                           body=None,
                                                                           *args)
```

```
exception boto.machinelearning.exceptions.InvalidInputException (status, reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.machinelearning.exceptions.LimitExceededException (status,
                                                                     reason,
                                                                     body=None,
                                                                     *args)
```

```
exception boto.machinelearning.exceptions.PredictorNotMountedException (status,
                                                                           rea-
                                                                           son,
                                                                           body=None,
                                                                           *args)
```

```
exception boto.machinelearning.exceptions.ResourceInUseException (status,
                                                                    reason,
                                                                    body=None,
                                                                    *args)
```

```
exception boto.machinelearning.exceptions.ResourceNotFoundException (status,
                                                                      reason,
                                                                      body=None,
                                                                      *args)
```

4.142 Opsworks

4.142.1 boto.opsworks

```
boto.opsworks.connect_to_region (region_name, **kw_params)
```

```
boto.opsworks.regions ()
```

Get all available regions for the Amazon OpsWorks service.

Return type list

Returns A list of boto.regioninfo.RegionInfo

4.142.2 boto.opsworks.layer1

class boto.opsworks.layer1.OpsWorksConnection (**kwargs)

AWS OpsWorks Welcome to the AWS OpsWorks API Reference . This guide provides descriptions, syntax, and usage examples about AWS OpsWorks actions and data types, including common parameters and error codes.

AWS OpsWorks is an application management service that provides an integrated experience for overseeing the complete application lifecycle. For information about this product, go to the [‘AWS OpsWorks’_](#) details page.

SDKs and CLI

The most common way to use the AWS OpsWorks API is by using the AWS Command Line Interface (CLI) or by using one of the AWS SDKs to implement applications in your preferred language. For more information, see:

- [‘AWS CLI’_](#)
- [‘AWS SDK for Java’_](#)
- [‘AWS SDK for .NET’_](#)
- [‘AWS SDK for PHP 2’_](#)
- [‘AWS SDK for Ruby’_](#)
- [‘AWS SDK for Node.js’_](#)
- [‘AWS SDK for Python\(Boto\)’_](#)

Endpoints

AWS OpsWorks supports only one endpoint, opsworks.us-east-1.amazonaws.com (HTTPS), so you must connect to that endpoint. You can then use the API to direct AWS OpsWorks to create stacks in any AWS Region.

Chef Versions

When you call CreateStack, CloneStack, or UpdateStack we recommend you use the *ConfigurationManager* parameter to specify the Chef version, 0.9, 11.4, or 11.10. The default value is currently 11.10. For more information, see [‘Chef Versions’_](#).

You can still specify Chef 0.9 for your stack, but new features are not available for Chef 0.9 stacks, and support is scheduled to end on July 24, 2014. We do not recommend using Chef 0.9 for new stacks, and we recommend migrating your existing Chef 0.9 stacks to Chef 11.10 as soon as possible.

APIVersion = '2013-02-18'

DefaultRegionEndpoint = 'opsworks.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of *boto.exception.JSONResponseError*

ServiceName = 'OpsWorks'

TargetPrefix = 'OpsWorks_20130218'

assign_instance (*instance_id*, *layer_ids*)

Assign a registered instance to a custom layer. You cannot use this action with instances that were created with AWS OpsWorks.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **instance_id** (*string*) – The instance ID.
- **layer_ids** (*list*) – The layer ID, which must correspond to a custom layer. You cannot assign a registered instance to a built-in layer.

assign_volume (*volume_id*, *instance_id=None*)

Assigns one of the stack's registered Amazon EBS volumes to a specified instance. The volume must first be registered with the stack by calling RegisterVolume. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **volume_id** (*string*) – The volume ID.
- **instance_id** (*string*) – The instance ID.

associate_elastic_ip (*elastic_ip*, *instance_id=None*)

Associates one of the stack's registered Elastic IP addresses with a specified instance. The address must first be registered with the stack by calling RegisterElasticIp. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **elastic_ip** (*string*) – The Elastic IP address.
- **instance_id** (*string*) – The instance ID.

attach_elastic_load_balancer (*elastic_load_balancer_name*, *layer_id*)

Attaches an Elastic Load Balancing load balancer to a specified layer. For more information, see [‘Elastic Load Balancing’](#).

You must create the Elastic Load Balancing instance separately, by using the Elastic Load Balancing console, API, or CLI. For more information, see [‘Elastic Load Balancing Developer Guide’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **elastic_load_balancer_name** (*string*) – The Elastic Load Balancing instance's name.
- **layer_id** (*string*) – The ID of the layer that the Elastic Load Balancing instance is to be attached to.

clone_stack (*source_stack_id*, *service_role_arn*, *name=None*, *region=None*, *vpc_id=None*, *attributes=None*, *default_instance_profile_arn=None*, *default_os=None*, *host_name_theme=None*, *default_availability_zone=None*, *default_subnet_id=None*, *custom_json=None*, *configuration_manager=None*, *chef_configuration=None*, *use_custom_cookbooks=None*, *use_opsworks_security_groups=None*, *custom_cookbooks_source=None*, *default_ssh_key_name=None*, *clone_permissions=None*, *clone_app_ids=None*, *default_root_device_type=None*)

Creates a clone of a specified stack. For more information, see [‘Clone a Stack’](#).

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **source_stack_id** (*string*) – The source stack ID.
- **name** (*string*) – The cloned stack name.

- **region** (*string*) – The cloned stack AWS region, such as “us-east-1”. For more information about AWS regions, see [‘Regions and Endpoints’_](#).
- **vpc_id** (*string*) – The ID of the VPC that the cloned stack is to be launched into. It must be in the specified region. All instances are launched into this VPC, and you cannot change the ID later.
- If your account supports EC2 Classic, the default value is no VPC.
- **If your account does not support EC2 Classic, the default value is** the default VPC for the specified region.

If the VPC ID corresponds to a default VPC and you have specified either the *DefaultAvailabilityZone* or the *DefaultSubnetId* parameter only, AWS OpsWorks infers the value of the other parameter. If you specify neither parameter, AWS OpsWorks sets these parameters to the first valid Availability Zone for the specified region and the corresponding default VPC subnet ID, respectively.

If you specify a nondefault VPC ID, note the following:

- **It must belong to a VPC in your account that is in the specified** region.
- You must specify a value for *DefaultSubnetId*.

For more information on how to use AWS OpsWorks with a VPC, see [‘Running a Stack in a VPC’_](#). For more information on default VPC and EC2 Classic, see [‘Supported Platforms’_](#).

Parameters

- **attributes** (*map*) – A list of stack attributes and values as key/value pairs to be added to the cloned stack.
- **service_role_arn** (*string*) –

The stack AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. If you create a stack by using the AWS OpsWorks console, it creates the role for you. You can obtain an existing stack’s IAM ARN programmatically by calling *DescribePermissions*. For more information about IAM ARNs, see [‘Using Identifiers’_](#).

You must set this parameter to a valid service role ARN or the action will fail; there is no default value. You can specify the source stack’s service role ARN, if you prefer, but you must do so explicitly.

Parameters

- **default_instance_profile_arn** (*string*) – The ARN of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [‘Using Identifiers’_](#).
- **default_os** (*string*) – The stack’s operating system, which must be set to one of the following.
 - **Standard operating systems: an Amazon Linux version such as** [‘Amazon Linux 2014.09’](#), [‘Ubuntu 12.04 LTS’](#), or [‘Ubuntu 14.04 LTS’](#).
 - **Custom AMIs: Custom.** You specify the custom AMI you want to use when you create instances.

The default option is the current Amazon Linux version.

Parameters hostname_theme (*string*) – The stack’s host name theme, with spaces are replaced by underscores. The theme is used to generate host names for the stack’s instances. By default, *HostnameTheme* is set to *Layer_Dependent*, which creates host names by appending integers to the layer’s short name. The other themes are:

- *Baked_Goods*

- *Clouds*
- *European_Cities*
- *Fruits*
- *Greek_Deities*
- *Legendary_Creatures_from_Japan*
- *Planets_and_Moons*
- *Roman_Deities*
- *Scottish_Islands*
- *US_Cities*
- *Wild_Cats*

To obtain a generated host name, call *GetHostNameSuggestion*, which returns a host name based on the current theme.

Parameters

- **default_availability_zone** (*string*) – The cloned stack’s default Availability Zone, which must be in the specified region. For more information, see [‘Regions and Endpoints’_](#). If you also specify a value for *DefaultSubnetId*, the subnet must be in the same zone. For more information, see the *VpcId* parameter description.
- **default_subnet_id** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the *VpcId* parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for *DefaultAvailabilityZone*, the subnet must be in that zone. For information on default values and when this parameter is required, see the *VpcId* parameter description.
- **custom_json** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as `“”` and `:`:

```
“{“key1”: “value1”, “key2”: “value2”,...}”
```

For more information on custom JSON, see [‘Use Custom JSON to Modify the Stack Configuration JSON’_](#)

Parameters

- **configuration_manager** (*dict*) – The configuration manager. When you clone a stack we recommend that you use the configuration manager to specify the Chef version, 0.9, 11.4, or 11.10. The default value is currently 11.4.
- **chef_configuration** (*dict*) – A *ChefConfiguration* object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [‘Create a New Stack’_](#).
- **use_custom_cookbooks** (*boolean*) – Whether to use custom cookbooks.
- **use_opsworks_security_groups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. With *UseOpsworksSecurityGroups* you can instead provide your own custom security groups. *UseOpsworksSecurityGroups* has the following settings:

- **True** - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it but you cannot delete the built-in security group.

- **False** - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see [‘Create a New Stack’_](#).

Parameters

- **custom_cookbooks_source** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [‘Creating Apps’_](#) or [‘Custom Recipes and Cookbooks’_](#).
- **default_ssh_key_name** (*string*) – A default SSH key for the stack instances. You can override this value when you create or update an instance.
- **clone_permissions** (*boolean*) – Whether to clone the source stack’s permissions.
- **clone_app_ids** (*list*) – A list of source stack app IDs to be included in the cloned stack.
- **default_root_device_type** (*string*) – The default root device type. This value is used by default for all instances in the cloned stack, but you can override it when you create an instance. For more information, see [‘Storage for the Root Device’_](#).

create_app (*stack_id*, *name*, *type*, *shortname=None*, *description=None*, *data_sources=None*, *app_source=None*, *domains=None*, *enable_ssl=None*, *ssl_configuration=None*, *attributes=None*, *environment=None*)

Creates an app for a specified stack. For more information, see [‘Creating Apps’_](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **stack_id** (*string*) – The stack ID.
- **shortname** (*string*) – The app’s short name.
- **name** (*string*) – The app name.
- **description** (*string*) – A description of the app.
- **data_sources** (*list*) – The app’s data source.
- **type** (*string*) – The app type. Each supported type is associated with a particular layer. For example, PHP applications are associated with a PHP layer. AWS OpsWorks deploys an application to those instances that are members of the corresponding layer.
- **app_source** (*dict*) – A *Source* object that specifies the app repository.
- **domains** (*list*) – The app virtual host settings, with multiple domains separated by commas. For example: `‘www.example.com, example.com’`
- **enable_ssl** (*boolean*) – Whether to enable SSL for the app.
- **ssl_configuration** (*dict*) – An *SslConfiguration* object with the SSL configuration.
- **attributes** (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.

- **environment** (*list*) –

An array of *EnvironmentVariable* objects that specify environment variables to be associated with the app. You can specify up to ten environment variables. After you deploy the app, these variables are defined on the associated app server instance.

This parameter is supported only by Chef 11.10 stacks. If you have specified one or more environment variables, you cannot modify the stack's Chef version.

create_deployment (*stack_id*, *command*, *app_id=None*, *instance_ids=None*, *comment=None*, *custom_json=None*)

Runs deployment or stack commands. For more information, see [‘Deploying Apps’](#) and [‘Run Stack Commands’](#).

Required Permissions: To use this action, an IAM user must have a Deploy or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **stack_id** (*string*) – The stack ID.
- **app_id** (*string*) – The app ID. This parameter is required for app deployments, but not for other deployment commands.
- **instance_ids** (*list*) – The instance IDs for the deployment targets.
- **command** (*dict*) – A *DeploymentCommand* object that specifies the deployment command and any associated arguments.
- **comment** (*string*) – A user-defined comment.
- **custom_json** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as “”:

```
“{“key1”: “value1”, “key2”: “value2”,...}”
```

For more information on custom JSON, see [‘Use Custom JSON to Modify the Stack Configuration JSON’](#).

create_instance (*stack_id*, *layer_ids*, *instance_type*, *auto_scaling_type=None*, *hostname=None*, *os=None*, *ami_id=None*, *ssh_key_name=None*, *availability_zone=None*, *virtualization_type=None*, *subnet_id=None*, *architecture=None*, *root_device_type=None*, *install_updates_on_boot=None*, *ebs_optimized=None*)

Creates an instance in a specified stack. For more information, see [‘Adding an Instance to a Layer’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **stack_id** (*string*) – The stack ID.
- **layer_ids** (*list*) – An array that contains the instance layer IDs.
- **instance_type** (*string*) – The instance type. AWS OpsWorks supports all instance types except Cluster Compute, Cluster GPU, and High Memory Cluster. For more information, see [‘Instance Families and Types’](#). The parameter values that you use to specify the various types are in the API Name column of the Available Instance Types table.
- **auto_scaling_type** (*string*) – For load-based or time-based instances, the type.

- **hostname** (*string*) – The instance host name.
- **os** (*string*) – The instance’s operating system, which must be set to one of the following.
- **Standard operating systems: an Amazon Linux version such as ‘Amazon Linux 2014.09’, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.**
- Custom AMIs: *Custom*

The default option is the current Amazon Linux version. If you set this parameter to *Custom*, you must use the `CreateInstance` action’s `AmiId` parameter to specify the custom AMI that you want to use. For more information on the standard operating systems, see [‘Operating Systems’_For more information on how to use custom AMIs with OpsWorks, see ‘Using Custom AMIs’_](#).

Parameters `ami_id` (*string*) –

A custom AMI ID to be used to create the instance. The AMI should be based on one of the standard AWS OpsWorks AMIs: Amazon Linux, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS. For more information, see [‘Instances’_](#).

If you specify a custom AMI, you must set `Os` to *Custom*.

Parameters

- **ssh_key_name** (*string*) – The instance SSH key name.
- **availability_zone** (*string*) – The instance Availability Zone. For more information, see [‘Regions and Endpoints’_](#).
- **virtualization_type** (*string*) – The instance’s virtualization type, *paravirtual* or *hvm*.
- **subnet_id** (*string*) – The ID of the instance’s subnet. If the stack is running in a VPC, you can use this parameter to override the stack’s default subnet ID value and direct AWS OpsWorks to launch the instance in a different subnet.
- **architecture** (*string*) – The instance architecture. The default option is *x86_64*. Instance types do not necessarily support both architectures. For a list of the architectures that are supported by the different instance types, see [‘Instance Families and Types’_](#).
- **root_device_type** (*string*) – The instance root device type. For more information, see [‘Storage for the Root Device’_](#).
- **install_updates_on_boot** (*boolean*) –

Whether to install operating system and package updates when the instance boots. The default value is *True*. To control when updates are installed, set this value to *False*. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

We strongly recommend using the default value of *True* to ensure that your instances have the latest security updates.

Parameters `ebs_optimized` (*boolean*) – Whether to create an Amazon EBS-optimized instance.

create_layer (*stack_id*, *type*, *name*, *shortname*, *attributes=None*, *custom_instance_profile_arn=None*, *custom_security_group_ids=None*, *packages=None*, *volume_configurations=None*, *enable_auto_healing=None*, *auto_assign_elastic_ips=None*, *auto_assign_public_ips=None*, *custom_recipes=None*, *install_updates_on_boot=None*, *use_ebs_optimized_instances=None*, *lifecycle_event_configuration=None*)

Creates a layer. For more information, see [‘How to Create a Layer’_](#).

You should use **CreateLayer** for noncustom layer types such as PHP App Server only if the stack does not have an existing layer of that type. A stack can have at most one instance of each noncustom layer; if you attempt to create a second instance, **CreateLayer** fails. A stack can have an arbitrary number of custom layers, so you can call **CreateLayer** as many times as you like for that layer type.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **stack_id** (*string*) – The layer stack ID.
- **type** (*string*) – The layer type. A stack cannot have more than one built-in layer of the same type. It can have any number of custom layers.
- **name** (*string*) – The layer name, which is used by the console.
- **shortname** (*string*) – The layer short name, which is used internally by AWS OpsWorks and by Chef recipes. The short name is also used as the name for the directory where your app files are installed. It can have a maximum of 200 characters, which are limited to the alphanumeric characters, ‘-’, ‘_’, and ‘.’.
- **attributes** (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.
- **custom_instance_profile_arn** (*string*) – The ARN of an IAM profile that to be used for the layer’s EC2 instances. For more information about IAM ARNs, see [‘Using Identifiers’](#).
- **custom_security_group_ids** (*list*) – An array containing the layer custom security group IDs.
- **packages** (*list*) – An array of *Package* objects that describe the layer packages.
- **volume_configurations** (*list*) – A *VolumeConfigurations* object that describes the layer’s Amazon EBS volumes.
- **enable_auto_healing** (*boolean*) – Whether to disable auto healing for the layer.
- **auto_assign_elastic_ips** (*boolean*) – Whether to automatically assign an [‘Elastic IP address’](#) to the layer’s instances. For more information, see [‘How to Edit a Layer’](#).
- **auto_assign_public_ips** (*boolean*) – For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer’s instances. For more information, see [‘How to Edit a Layer’](#).
- **custom_recipes** (*dict*) – A *LayerCustomRecipes* object that specifies the layer custom recipes.
- **install_updates_on_boot** (*boolean*) –

Whether to install operating system and package updates when the instance boots. The default value is *True*. To control when updates are installed, set this value to *False*. You must then update your instances manually by using `CreateDeployment` to run the *update_dependencies* stack command or manually running *yum* (Amazon Linux) or *apt-get* (Ubuntu) on the instances.

We strongly recommend using the default value of *True*, to ensure that your instances have the latest security updates.

Parameters

- **use_ebs_optimized_instances** (*boolean*) – Whether to use Amazon EBS-optimized instances.
- **lifecycle_event_configuration** (*dict*) – A LifecycleEventConfiguration object that you can use to configure the Shutdown event to specify an execution timeout and enable or disable Elastic Load Balancer connection draining.

create_stack (*name*, *region*, *service_role_arn*, *default_instance_profile_arn*, *vpc_id=None*, *attributes=None*, *default_os=None*, *hostname_theme=None*, *default_availability_zone=None*, *default_subnet_id=None*, *custom_json=None*, *configuration_manager=None*, *chef_configuration=None*, *use_custom_cookbooks=None*, *use_opsworks_security_groups=None*, *custom_cookbooks_source=None*, *default_ssh_key_name=None*, *default_root_device_type=None*)

Creates a new stack. For more information, see [‘Create a New Stack’](#).

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **name** (*string*) – The stack name.
- **region** (*string*) – The stack AWS region, such as “us-east-1”. For more information about Amazon regions, see [‘Regions and Endpoints’](#).
- **vpc_id** (*string*) – The ID of the VPC that the stack is to be launched into. It must be in the specified region. All instances are launched into this VPC, and you cannot change the ID later.
 - If your account supports EC2 Classic, the default value is no VPC.
 - **If your account does not support EC2 Classic, the default value is the default VPC for the specified region.**

If the VPC ID corresponds to a default VPC and you have specified either the *DefaultAvailabilityZone* or the *DefaultSubnetId* parameter only, AWS OpsWorks infers the value of the other parameter. If you specify neither parameter, AWS OpsWorks sets these parameters to the first valid Availability Zone for the specified region and the corresponding default VPC subnet ID, respectively.

If you specify a nondefault VPC ID, note the following:

- **It must belong to a VPC in your account that is in the specified** region.
- You must specify a value for *DefaultSubnetId*.

For more information on how to use AWS OpsWorks with a VPC, see [‘Running a Stack in a VPC’](#). For more information on default VPC and EC2 Classic, see [‘Supported Platforms’](#).

Parameters

- **attributes** (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.
- **service_role_arn** (*string*) – The stack AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. For more information about IAM ARNs, see [‘Using Identifiers’](#).
- **default_instance_profile_arn** (*string*) – The ARN of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [‘Using Identifiers’](#).

- **default_os** (*string*) – The stack’s operating system, which must be set to one of the following.
- **Standard operating systems: an Amazon Linux version such as ‘Amazon Linux 2014.09’, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.**
- **Custom AMIs: Custom.** You specify the custom AMI you want to use when you create instances.

The default option is the current Amazon Linux version.

Parameters hostname_theme (*string*) – The stack’s host name theme, with spaces are replaced by underscores. The theme is used to generate host names for the stack’s instances. By default, *HostnameTheme* is set to *Layer_Dependent*, which creates host names by appending integers to the layer’s short name. The other themes are:

- *Baked_Goods*
- *Clouds*
- *European_Cities*
- *Fruits*
- *Greek_Deities*
- *Legendary_Creatures_from_Japan*
- *Planets_and_Moons*
- *Roman_Deities*
- *Scottish_Islands*
- *US_Cities*
- *Wild_Cats*

To obtain a generated host name, call *GetHostNameSuggestion*, which returns a host name based on the current theme.

Parameters

- **default_availability_zone** (*string*) – The stack’s default Availability Zone, which must be in the specified region. For more information, see **‘Regions and Endpoints’**. If you also specify a value for *DefaultSubnetId*, the subnet must be in the same zone. For more information, see the *VpcId* parameter description.
- **default_subnet_id** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the *VpcId* parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for *DefaultAvailabilityZone*, the subnet must be in that zone. For information on default values and when this parameter is required, see the *VpcId* parameter description.
- **custom_json** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as “”:

“{“key1”: “value1”, “key2”: “value2”,...}”

For more information on custom JSON, see ‘Use Custom JSON to Modify the Stack Configuration JSON’.

Parameters

- **configuration_manager** (*dict*) – The configuration manager. When you clone a stack we recommend that you use the configuration manager to specify the Chef version, 0.9, 11.4, or 11.10. The default value is currently 11.4.

- **chef_configuration** (*dict*) – A *ChefConfiguration* object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see [‘Create a New Stack’_](#).
- **use_custom_cookbooks** (*boolean*) – Whether the stack uses custom cookbooks.
- **use_opsworks_security_groups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. With *UseOpsworksSecurityGroups* you can instead provide your own custom security groups. *UseOpsworksSecurityGroups* has the following settings:

- **True** - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it but you cannot delete the built-in security group.
- **False** - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see [‘Create a New Stack’_](#).

Parameters

- **custom_cookbooks_source** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see [‘Creating Apps’_](#) or [‘Custom Recipes and Cookbooks’_](#).
- **default_ssh_key_name** (*string*) – A default SSH key for the stack instances. You can override this value when you create or update an instance.
- **default_root_device_type** (*string*) – The default root device type. This value is used by default for all instances in the stack, but you can override it when you create an instance. The default option is *instance-store*. For more information, see [‘Storage for the Root Device’_](#).

create_user_profile (*iam_user_arn*, *ssh_username=None*, *ssh_public_key=None*, *allow_self_management=None*)

Creates a new user profile.

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **iam_user_arn** (*string*) – The user’s IAM ARN.
- **ssh_username** (*string*) – The user’s SSH user name. The allowable characters are [a-z], [A-Z], [0-9], ‘-’, and ‘_’. If the specified name includes other punctuation marks, AWS OpsWorks removes them. For example, *my.name* will be changed to *myname*. If you do not specify an SSH user name, AWS OpsWorks generates one from the IAM user name.
- **ssh_public_key** (*string*) – The user’s public SSH key.
- **allow_self_management** (*boolean*) – Whether users can specify their own SSH public key through the My Settings page. For more information, see [‘Setting an IAM User’s Public SSH Key’_](#).

delete_app (*app_id*)

Deletes a specified app.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters `app_id` (*string*) – The app ID.

`delete_instance` (*instance_id*, *delete_elastic_ip=None*, *delete_volumes=None*)

Deletes a specified instance, which terminates the associated Amazon EC2 instance. You must stop an instance before you can delete it.

For more information, see [‘Deleting Instances’_](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- `instance_id` (*string*) – The instance ID.
- `delete_elastic_ip` (*boolean*) – Whether to delete the instance Elastic IP address.
- `delete_volumes` (*boolean*) – Whether to delete the instance’s Amazon EBS volumes.

`delete_layer` (*layer_id*)

Deletes a specified layer. You must first stop and then delete all associated instances or unassign registered instances. For more information, see [‘How to Delete a Layer’_](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters `layer_id` (*string*) – The layer ID.

`delete_stack` (*stack_id*)

Deletes a specified stack. You must first delete all instances, layers, and apps or deregister registered instances. For more information, see [‘Shut Down a Stack’_](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters `stack_id` (*string*) – The stack ID.

`delete_user_profile` (*iam_user_arn*)

Deletes a user profile.

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters `iam_user_arn` (*string*) – The user’s IAM ARN.

`deregister_elastic_ip` (*elastic_ip*)

Deregisters a specified Elastic IP address. The address can then be registered by another stack. For more information, see [‘Resource Management’_](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters `elastic_ip` (*string*) – The Elastic IP address.

`deregister_instance` (*instance_id*)

Deregister a registered Amazon EC2 or on-premises instance. This action removes the instance from the

stack and returns it to your control. This action can not be used with instances that were created with AWS OpsWorks.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `instance_id` (*string*) – The instance ID.

deregister_rds_db_instance (*rds_db_instance_arn*)

Deregisters an Amazon RDS instance.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `rds_db_instance_arn` (*string*) – The Amazon RDS instance’s ARN.

deregister_volume (*volume_id*)

Deregisters an Amazon EBS volume. The volume can then be registered by another stack. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `volume_id` (*string*) – The volume ID.

describe_apps (*stack_id=None, app_ids=None*)

Requests a description of a specified set of apps.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `stack_id` (*string*) – The app stack ID. If you use this parameter, `DescribeApps` returns a description of the apps in the specified stack.
- `app_ids` (*list*) – An array of app IDs for the apps to be described. If you use this parameter, `DescribeApps` returns a description of the specified apps. Otherwise, it returns a description of every app.

describe_commands (*deployment_id=None, instance_id=None, command_ids=None*)

Describes the results of specified commands.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `deployment_id` (*string*) – The deployment ID. If you include this parameter, `DescribeCommands` returns a description of the commands associated with the specified deployment.
- `instance_id` (*string*) – The instance ID. If you include this parameter, `DescribeCommands` returns a description of the commands associated with the specified instance.

- **command_ids** (*list*) – An array of command IDs. If you include this parameter, *DescribeCommands* returns a description of the specified commands. Otherwise, it returns a description of every command.

describe_deployments (*stack_id=None, app_id=None, deployment_ids=None*)

Requests a description of a specified set of deployments.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions‘](#).

Parameters

- **stack_id** (*string*) – The stack ID. If you include this parameter, *DescribeDeployments* returns a description of the commands associated with the specified stack.
- **app_id** (*string*) – The app ID. If you include this parameter, *DescribeDeployments* returns a description of the commands associated with the specified app.
- **deployment_ids** (*list*) – An array of deployment IDs to be described. If you include this parameter, *DescribeDeployments* returns a description of the specified deployments. Otherwise, it returns a description of every deployment.

describe_elastic_ips (*instance_id=None, stack_id=None, ips=None*)

Describes [‘Elastic IP addresses‘](#).

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions‘](#).

Parameters

- **instance_id** (*string*) – The instance ID. If you include this parameter, *DescribeElasticIps* returns a description of the Elastic IP addresses associated with the specified instance.
- **stack_id** (*string*) – A stack ID. If you include this parameter, *DescribeElasticIps* returns a description of the Elastic IP addresses that are registered with the specified stack.
- **ips** (*list*) – An array of Elastic IP addresses to be described. If you include this parameter, *DescribeElasticIps* returns a description of the specified Elastic IP addresses. Otherwise, it returns a description of every Elastic IP address.

describe_elastic_load_balancers (*stack_id=None, layer_ids=None*)

Describes a stack’s Elastic Load Balancing instances.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions‘](#).

Parameters

- **stack_id** (*string*) – A stack ID. The action describes the stack’s Elastic Load Balancing instances.

- **layer_ids** (*list*) – A list of layer IDs. The action describes the Elastic Load Balancing instances for the specified layers.

describe_instances (*stack_id=None, layer_id=None, instance_ids=None*)

Requests a description of a set of instances.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **stack_id** (*string*) – A stack ID. If you use this parameter, *DescribeInstances* returns descriptions of the instances associated with the specified stack.
- **layer_id** (*string*) – A layer ID. If you use this parameter, *DescribeInstances* returns descriptions of the instances associated with the specified layer.
- **instance_ids** (*list*) – An array of instance IDs to be described. If you use this parameter, *DescribeInstances* returns a description of the specified instances. Otherwise, it returns a description of every instance.

describe_layers (*stack_id=None, layer_ids=None*)

Requests a description of one or more layers in a specified stack.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **stack_id** (*string*) – The stack ID.
- **layer_ids** (*list*) – An array of layer IDs that specify the layers to be described. If you omit this parameter, *DescribeLayers* returns a description of every layer in the specified stack.

describe_load_based_auto_scaling (*layer_ids*)

Describes load-based auto scaling configurations for specified layers.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters **layer_ids** (*list*) – An array of layer IDs.

describe_my_user_profile ()

Describes a user’s SSH information.

Required Permissions: To use this action, an IAM user must have self-management enabled or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

describe_permissions (*iam_user_arn=None, stack_id=None*)

Describes the permissions for a specified stack.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **iam_user_arn** (*string*) – The user’s IAM ARN. For more information about IAM ARNs, see [‘Using Identifiers’_](#).
- **stack_id** (*string*) – The stack ID.

describe_raid_arrays (*instance_id=None, stack_id=None, raid_array_ids=None*)

Describe an instance’s RAID arrays.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **instance_id** (*string*) – The instance ID. If you use this parameter, *DescribeRaidArrays* returns descriptions of the RAID arrays associated with the specified instance.
- **stack_id** (*string*) – The stack ID.
- **raid_array_ids** (*list*) – An array of RAID array IDs. If you use this parameter, *DescribeRaidArrays* returns descriptions of the specified arrays. Otherwise, it returns a description of every array.

describe_rds_db_instances (*stack_id, rds_db_instance_arns=None*)

Describes Amazon RDS instances.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **stack_id** (*string*) – The stack ID that the instances are registered with. The operation returns descriptions of all registered Amazon RDS instances.
- **rds_db_instance_arns** (*list*) – An array containing the ARNs of the instances to be described.

describe_service_errors (*stack_id=None, instance_id=None, service_error_ids=None*)

Describes AWS OpsWorks service errors.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **stack_id** (*string*) – The stack ID. If you use this parameter, *DescribeServiceErrors* returns descriptions of the errors associated with the specified stack.
- **instance_id** (*string*) – The instance ID. If you use this parameter, *DescribeServiceErrors* returns descriptions of the errors associated with the specified instance.
- **service_error_ids** (*list*) – An array of service error IDs. If you use this parameter, *DescribeServiceErrors* returns descriptions of the specified errors. Otherwise, it returns a description of every error.

describe_stack_provisioning_parameters (*stack_id*)

Requests a description of a stack’s provisioning parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `stack_id` (*string*) – The stack ID

describe_stack_summary (*stack_id*)

Describes the number of layers and apps in a specified stack, and the number of instances in each state, such as *running_setup* or *online*.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `stack_id` (*string*) – The stack ID.

describe_stacks (*stack_ids=None*)

Requests a description of one or more stacks.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `stack_ids` (*list*) – An array of stack IDs that specify the stacks to be described. If you omit this parameter, *DescribeStacks* returns a description of every stack.

describe_time_based_auto_scaling (*instance_ids*)

Describes time-based auto scaling configurations for specified instances.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `instance_ids` (*list*) – An array of instance IDs.

describe_user_profiles (*iam_user_arns=None*)

Describe specified users.

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `iam_user_arns` (*list*) – An array of IAM user ARNs that identify the users to be described.

describe_volumes (*instance_id=None, stack_id=None, raid_array_id=None, volume_ids=None*)

Describes an instance’s Amazon EBS volumes.

You must specify at least one of the parameters.

Required Permissions: To use this action, an IAM user must have a Show, Deploy, or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **instance_id** (*string*) – The instance ID. If you use this parameter, *DescribeVolumes* returns descriptions of the volumes associated with the specified instance.
- **stack_id** (*string*) – A stack ID. The action describes the stack’s registered Amazon EBS volumes.
- **raid_array_id** (*string*) – The RAID array ID. If you use this parameter,

DescribeVolumes returns descriptions of the volumes associated with the specified RAID array.

- **volume_ids** (*list*) – An array of volume IDs. If you use this parameter, *DescribeVolumes* returns descriptions of the specified volumes. Otherwise, it returns a description of every volume.

detach_elastic_load_balancer (*elastic_load_balancer_name, layer_id*)

Detaches a specified Elastic Load Balancing instance from its layer.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **elastic_load_balancer_name** (*string*) – The Elastic Load Balancing instance’s name.
- **layer_id** (*string*) – The ID of the layer that the Elastic Load Balancing instance is attached to.

disassociate_elastic_ip (*elastic_ip*)

Disassociates an Elastic IP address from its instance. The address remains registered with the stack. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters **elastic_ip** (*string*) – The Elastic IP address.

get_hostname_suggestion (*layer_id*)

Gets a generated host name for the specified layer, based on the current host name theme.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters **layer_id** (*string*) – The layer ID.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

reboot_instance (*instance_id*)

Reboots a specified instance. For more information, see [‘Starting, Stopping, and Rebooting Instances’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters **instance_id** (*string*) – The instance ID.

register_elastic_ip (*elastic_ip, stack_id*)

Registers an Elastic IP address with a specified stack. An address can be registered with only one stack at a time. If the address is already registered, you must first deregister it by calling `DeregisterElasticIp`. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- **elastic_ip** (*string*) – The Elastic IP address.

- `stack_id` (*string*) – The stack ID.

`register_instance` (*stack_id*, *hostname=None*, *public_ip=None*, *private_ip=None*, *rsa_public_key=None*, *rsa_public_key_fingerprint=None*, *instance_identity=None*)

Registers instances with a specified stack that were created outside of AWS OpsWorks.

We do not recommend using this action to register instances. The complete registration operation has two primary steps, installing the AWS OpsWorks agent on the instance and registering the instance with the stack. `RegisterInstance` handles only the second step. You should instead use the AWS CLI `register` command, which performs the entire registration operation.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `stack_id` (*string*) – The ID of the stack that the instance is to be registered with.
- `hostname` (*string*) – The instance’s hostname.
- `public_ip` (*string*) – The instance’s public IP address.
- `private_ip` (*string*) – The instance’s private IP address.
- `rsa_public_key` (*string*) – The instances public RSA key. This key is used to encrypt communication between the instance and the service.
- `rsa_public_key_fingerprint` (*string*) – The instances public RSA key fingerprint.
- `instance_identity` (*dict*) – An InstanceIdentity object that contains the instance’s identity.

`register_rds_db_instance` (*stack_id*, *rds_db_instance_arn*, *db_user*, *db_password*)

Registers an Amazon RDS instance with a stack.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `stack_id` (*string*) – The stack ID.
- `rds_db_instance_arn` (*string*) – The Amazon RDS instance’s ARN.
- `db_user` (*string*) – The database’s master user name.
- `db_password` (*string*) – The database password.

`register_volume` (*stack_id*, *ec_2_volume_id=None*)

Registers an Amazon EBS volume with a specified stack. A volume can be registered with only one stack at a time. If the volume is already registered, you must first deregister it by calling `DeregisterVolume`. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `ec_2_volume_id` (*string*) – The Amazon EBS volume ID.
- `stack_id` (*string*) – The stack ID.

`set_load_based_auto_scaling` (*layer_id*, *enable=None*, *up_scaling=None*,
down_scaling=None)

Specify the load-based auto scaling configuration for a specified layer. For more information, see **‘Managing Load with Time-based and Load-based Instances’**.

To use load-based auto scaling, you must create a set of load-based auto scaling instances. Load-based auto scaling operates only on the instances from that set, so you must ensure that you have created enough instances to handle the maximum anticipated load.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters

- **layer_id** (*string*) – The layer ID.
- **enable** (*boolean*) – Enables load-based auto scaling for the layer.
- **up_scaling** (*dict*) – An *AutoScalingThresholds* object with the upscaling threshold configuration. If the load exceeds these thresholds for a specified amount of time, AWS OpsWorks starts a specified number of instances.
- **down_scaling** (*dict*) – An *AutoScalingThresholds* object with the down-scaling threshold configuration. If the load falls below these thresholds for a specified amount of time, AWS OpsWorks stops a specified number of instances.

`set_permission` (*stack_id*, *iam_user_arn*, *allow_ssh=None*, *allow_sudo=None*, *level=None*)

Specifies a user’s permissions. For more information, see **‘Security and Permissions’**.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters

- **stack_id** (*string*) – The stack ID.
- **iam_user_arn** (*string*) – The user’s IAM ARN.
- **allow_ssh** (*boolean*) – The user is allowed to use SSH to communicate with the instance.
- **allow_sudo** (*boolean*) – The user is allowed to use **sudo** to elevate privileges.
- **level** (*string*) – The user’s permission level, which must be set to one of the following strings. You cannot set your own permissions level.

- *deny*
- *show*
- *deploy*
- *manage*
- *iam_only*

For more information on the permissions associated with these levels, see **‘Managing User Permissions’**.

`set_time_based_auto_scaling` (*instance_id*, *auto_scaling_schedule=None*)

Specify the time-based auto scaling configuration for a specified instance. For more information, see **‘Managing Load with Time-based and Load-based Instances’**.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters

- `instance_id` (*string*) – The instance ID.
- `auto_scaling_schedule` (*dict*) – An *AutoScalingSchedule* with the instance schedule.

start_instance (*instance_id*)

Starts a specified instance. For more information, see **‘Starting, Stopping, and Rebooting Instances’**.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `instance_id` (*string*) – The instance ID.

start_stack (*stack_id*)

Starts a stack’s instances.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `stack_id` (*string*) – The stack ID.

stop_instance (*instance_id*)

Stops a specified instance. When you stop a standard instance, the data disappears and must be reinstalled when you restart the instance. You can stop an Amazon EBS-backed instance without losing data. For more information, see **‘Starting, Stopping, and Rebooting Instances’**.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `instance_id` (*string*) – The instance ID.

stop_stack (*stack_id*)

Stops a specified stack.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `stack_id` (*string*) – The stack ID.

unassign_instance (*instance_id*)

Unassigns a registered instance from all of it’s layers. The instance remains in the stack as an unassigned instance and can be assigned to another layer, as needed. You cannot use this action with instances that were created with AWS OpsWorks.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `instance_id` (*string*) – The instance ID.

unassign_volume (*volume_id*)

Unassigns an assigned Amazon EBS volume. The volume remains registered with the stack. For more information, see **‘Resource Management’**.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’**.

Parameters `volume_id` (*string*) – The volume ID.

`update_app` (*app_id*, *name=None*, *description=None*, *data_sources=None*, *type=None*, *app_source=None*, *domains=None*, *enable_ssl=None*, *ssl_configuration=None*, *attributes=None*, *environment=None*)

Updates a specified app.

Required Permissions: To use this action, an IAM user must have a Deploy or Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `app_id` (*string*) – The app ID.
- `name` (*string*) – The app name.
- `description` (*string*) – A description of the app.
- `data_sources` (*list*) – The app’s data sources.
- `type` (*string*) – The app type.
- `app_source` (*dict*) – A *Source* object that specifies the app repository.
- `domains` (*list*) – The app’s virtual host settings, with multiple domains separated by commas. For example: `‘www.example.com, example.com’`
- `enable_ssl` (*boolean*) – Whether SSL is enabled for the app.
- `ssl_configuration` (*dict*) – An *SslConfiguration* object with the SSL configuration.
- `attributes` (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.
- `environment` (*list*) –

An array of *EnvironmentVariable* objects that specify environment variables to be associated with the app. You can specify up to ten environment variables. After you deploy the app, these variables are defined on the associated app server instances.

This parameter is supported only by Chef 11.10 stacks. If you have specified one or more environment variables, you cannot modify the stack’s Chef version.

`update_elastic_ip` (*elastic_ip*, *name=None*)

Updates a registered Elastic IP address’s name. For more information, see [‘Resource Management’](#).

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `elastic_ip` (*string*) – The address.
- `name` (*string*) – The new name.

`update_instance` (*instance_id*, *layer_ids=None*, *instance_type=None*, *auto_scaling_type=None*, *hostname=None*, *os=None*, *ami_id=None*, *ssh_key_name=None*, *architecture=None*, *install_updates_on_boot=None*, *ebs_optimized=None*)

Updates a specified instance.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `instance_id` (*string*) – The instance ID.

- **layer_ids** (*list*) – The instance’s layer IDs.
- **instance_type** (*string*) – The instance type. AWS OpsWorks supports all instance types except Cluster Compute, Cluster GPU, and High Memory Cluster. For more information, see [‘Instance Families and Types’](#). The parameter values that you use to specify the various types are in the API Name column of the Available Instance Types table.
- **auto_scaling_type** (*string*) – For load-based or time-based instances, the type.
- **hostname** (*string*) – The instance host name.
- **os** (*string*) – The instance’s operating system, which must be set to one of the following.
 - **Standard operating systems: An Amazon Linux version such as ‘Amazon Linux 2014.09’, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS.**
 - Custom AMIs: *Custom*

The default option is the current Amazon Linux version, such as ‘Amazon Linux 2014.09’. If you set this parameter to *Custom*, you must use the CreateInstance action’s `AmiId` parameter to specify the custom AMI that you want to use. For more information on the standard operating systems, see [‘Operating Systems’](#). For more information on how to use custom AMIs with OpsWorks, see [‘Using Custom AMIs’](#).

Parameters `ami_id` (*string*) –

A custom AMI ID to be used to create the instance. The AMI should be based on one of the standard AWS OpsWorks AMIs: Amazon Linux, Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS. For more information, see [‘Instances’](#).

If you specify a custom AMI, you must set `Os` to *Custom*.

Parameters

- **ssh_key_name** (*string*) – The instance SSH key name.
- **architecture** (*string*) – The instance architecture. Instance types do not necessarily support both architectures. For a list of the architectures that are supported by the different instance types, see [‘Instance Families and Types’](#).
- **install_updates_on_boot** (*boolean*) –

Whether to install operating system and package updates when the instance boots. The default value is *True*. To control when updates are installed, set this value to *False*. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

We strongly recommend using the default value of *True*, to ensure that your instances have the latest security updates.

Parameters `ebs_optimized` (*boolean*) – Whether this is an Amazon EBS-optimized instance.

update_layer (*layer_id*, *name=None*, *shortname=None*, *attributes=None*, *custom_instance_profile_arn=None*, *custom_security_group_ids=None*, *packages=None*, *volume_configurations=None*, *enable_auto_healing=None*, *auto_assign_elastic_ips=None*, *auto_assign_public_ips=None*, *custom_recipes=None*, *install_updates_on_boot=None*, *use_ebs_optimized_instances=None*, *lifecycle_event_configuration=None*)

Updates a specified layer.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’_](#).

Parameters

- **layer_id** (*string*) – The layer ID.
- **name** (*string*) – The layer name, which is used by the console.
- **shortname** (*string*) – The layer short name, which is used internally by AWS OpsWorks and by Chef. The short name is also used as the name for the directory where your app files are installed. It can have a maximum of 200 characters and must be in the following format: `/A[a-z0-9-_.]+Z/`.
- **attributes** (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.
- **custom_instance_profile_arn** (*string*) – The ARN of an IAM profile to be used for all of the layer’s EC2 instances. For more information about IAM ARNs, see [‘Using Identifiers’_](#).
- **custom_security_group_ids** (*list*) – An array containing the layer’s custom security group IDs.
- **packages** (*list*) – An array of *Package* objects that describe the layer’s packages.
- **volume_configurations** (*list*) – A *VolumeConfigurations* object that describes the layer’s Amazon EBS volumes.
- **enable_auto_healing** (*boolean*) – Whether to disable auto healing for the layer.
- **auto_assign_elastic_ips** (*boolean*) – Whether to automatically assign an [‘Elastic IP address’_](#) to the layer’s instances. For more information, see [‘How to Edit a Layer’_](#).
- **auto_assign_public_ips** (*boolean*) – For stacks that are running in a VPC, whether to automatically assign a public IP address to the layer’s instances. For more information, see [‘How to Edit a Layer’_](#).
- **custom_recipes** (*dict*) – A *LayerCustomRecipes* object that specifies the layer’s custom recipes.
- **install_updates_on_boot** (*boolean*) –

Whether to install operating system and package updates when the instance boots. The default value is *True*. To control when updates are installed, set this value to *False*. You must then update your instances manually by using `CreateDeployment` to run the `update_dependencies` stack command or manually running `yum` (Amazon Linux) or `apt-get` (Ubuntu) on the instances.

We strongly recommend using the default value of *True*, to ensure that your instances have the latest security updates.

Parameters

- **use_ebs_optimized_instances** (*boolean*) – Whether to use Amazon EBS-optimized instances.
- **lifecycle_event_configuration** (*dict*) –

update_my_user_profile (*ssh_public_key=None*)

Updates a user’s SSH public key.

Required Permissions: To use this action, an IAM user must have self-management enabled or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters `ssh_public_key` (*string*) – The user’s SSH public key.

`update_rds_db_instance` (*rds_db_instance_arn*, *db_user=None*, *db_password=None*)

Updates an Amazon RDS instance.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `rds_db_instance_arn` (*string*) – The Amazon RDS instance’s ARN.
- `db_user` (*string*) – The master user name.
- `db_password` (*string*) – The database password.

`update_stack` (*stack_id*, *name=None*, *attributes=None*, *service_role_arn=None*, *default_instance_profile_arn=None*, *default_os=None*, *hostname_theme=None*, *default_availability_zone=None*, *default_subnet_id=None*, *custom_json=None*, *configuration_manager=None*, *chef_configuration=None*, *use_custom_cookbooks=None*, *custom_cookbooks_source=None*, *default_ssh_key_name=None*, *default_root_device_type=None*, *use_opsworks_security_groups=None*)

Updates a specified stack.

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see [‘Managing User Permissions’](#).

Parameters

- `stack_id` (*string*) – The stack ID.
- `name` (*string*) – The stack’s new name.
- `attributes` (*map*) – One or more user-defined key/value pairs to be added to the stack attributes.
- `service_role_arn` (*string*) –

The stack AWS Identity and Access Management (IAM) role, which allows AWS OpsWorks to work with AWS resources on your behalf. You must set this parameter to the Amazon Resource Name (ARN) for an existing IAM role. For more information about IAM ARNs, see [‘Using Identifiers’](#).

You must set this parameter to a valid service role ARN or the action will fail; there is no default value. You can specify the stack’s current service role ARN, if you prefer, but you must do so explicitly.

Parameters

- `default_instance_profile_arn` (*string*) – The ARN of an IAM profile that is the default profile for all of the stack’s EC2 instances. For more information about IAM ARNs, see [‘Using Identifiers’](#).
- `default_os` (*string*) – The stack’s operating system, which must be set to one of the following.
 - **Standard operating systems: an Amazon Linux version such as ‘Amazon Linux 2014.09’, ‘Ubuntu 12.04 LTS’, or ‘Ubuntu 14.04 LTS’.**
 - **Custom AMIs: Custom.** You specify the custom AMI you want to use when you create instances.

The default option is the current Amazon Linux version.

Parameters `hostname_theme` (*string*) – The stack’s new host name theme, with spaces are replaced by underscores. The theme is used to generate host names for the stack’s instances. By default, `HostnameTheme` is set to `Layer_Dependent`, which creates host names by appending integers to the layer’s short name. The other themes are:

- `Baked_Goods`
- `Clouds`
- `European_Cities`
- `Fruits`
- `Greek_Deities`
- `Legendary_Creatures_from_Japan`
- `Planets_and_Moons`
- `Roman_Deities`
- `Scottish_Islands`
- `US_Cities`
- `Wild_Cats`

To obtain a generated host name, call `GetHostNameSuggestion`, which returns a host name based on the current theme.

Parameters

- **`default_availability_zone`** (*string*) – The stack’s default Availability Zone, which must be in the specified region. For more information, see **‘Regions and Endpoints’**. If you also specify a value for `DefaultSubnetId`, the subnet must be in the same zone. For more information, see `CreateStack`.
- **`default_subnet_id`** (*string*) – The stack’s default VPC subnet ID. This parameter is required if you specify a value for the `VpcId` parameter. All instances are launched into this subnet unless you specify otherwise when you create the instance. If you also specify a value for `DefaultAvailabilityZone`, the subnet must be in that zone. For information on default values and when this parameter is required, see the `VpcId` parameter description.
- **`custom_json`** (*string*) – A string that contains user-defined, custom JSON. It is used to override the corresponding default stack configuration JSON values. The string should be in the following format and must escape characters such as `”`:

```
{“key1”: “value1”, “key2”: “value2”,...}
```

For more information on custom JSON, see **‘Use Custom JSON to Modify the Stack Configuration JSON’.**

Parameters

- **`configuration_manager`** (*dict*) – The configuration manager. When you clone a stack we recommend that you use the configuration manager to specify the Chef version, 0.9, 11.4, or 11.10. The default value is currently 11.4.
- **`chef_configuration`** (*dict*) – A `ChefConfiguration` object that specifies whether to enable Berkshelf and the Berkshelf version on Chef 11.10 stacks. For more information, see **‘Create a New Stack’**.
- **`use_custom_cookbooks`** (*boolean*) – Whether the stack uses custom cookbooks.
- **`custom_cookbooks_source`** (*dict*) – Contains the information required to retrieve an app or cookbook from a repository. For more information, see

‘Creating Apps’_ or ‘Custom Recipes and Cookbooks’_.

- **default_ssh_key_name** (*string*) – A default SSH key for the stack instances. You can override this value when you create or update an instance.
- **default_root_device_type** (*string*) – The default root device type. This value is used by default for all instances in the stack, but you can override it when you create an instance. For more information, see **‘Storage for the Root Device’_.**
- **use_opsworks_security_groups** (*boolean*) – Whether to associate the AWS OpsWorks built-in security groups with the stack’s layers.

AWS OpsWorks provides a standard set of built-in security groups, one for each layer, which are associated with layers by default. *UseOpsworksSecurityGroups* allows you to instead provide your own custom security groups. *UseOpsworksSecurityGroups* has the following settings:

- **True** - AWS OpsWorks automatically associates the appropriate built-in security group with each layer (default setting). You can associate additional security groups with a layer after you create it but you cannot delete the built-in security group.
- **False** - AWS OpsWorks does not associate built-in security groups with layers. You must create appropriate EC2 security groups and associate a security group with each layer that you create. However, you can still manually associate a built-in security group with a layer on creation; custom security groups are required only for those layers that need custom settings.

For more information, see **‘Create a New Stack’_.**

update_user_profile (*iam_user_arn*, *ssh_username=None*, *ssh_public_key=None*, *allow_self_management=None*)

Updates a specified user profile.

Required Permissions: To use this action, an IAM user must have an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’_.**

Parameters

- **iam_user_arn** (*string*) – The user IAM ARN.
- **ssh_username** (*string*) – The user’s SSH user name. The allowable characters are [a-z], [A-Z], [0-9], ‘-’, and ‘_’. If the specified name includes other punctuation marks, AWS OpsWorks removes them. For example, *my.name* will be changed to *myname*. If you do not specify an SSH user name, AWS OpsWorks generates one from the IAM user name.
- **ssh_public_key** (*string*) – The user’s new SSH public key.
- **allow_self_management** (*boolean*) – Whether users can specify their own SSH public key through the My Settings page. For more information, see **‘Managing User Permissions’_.**

update_volume (*volume_id*, *name=None*, *mount_point=None*)

Updates an Amazon EBS volume’s name or mount point. For more information, see **‘Resource Management’_.**

Required Permissions: To use this action, an IAM user must have a Manage permissions level for the stack, or an attached policy that explicitly grants permissions. For more information on user permissions, see **‘Managing User Permissions’_.**

Parameters

- **volume_id** (*string*) – The volume ID.
- **name** (*string*) – The new name.
- **mount_point** (*string*) – The new mount point.

4.142.3 boto.opsworks.exceptions

exception `boto.opsworks.exceptions.ResourceNotFoundException` (*status*, *reason*,
body=None,
**args*)

exception `boto.opsworks.exceptions.ValidationException` (*status*, *reason*, *body=None*,
**args*)

4.143 RDS2

4.143.1 boto.rds2

`boto.rds2.connect_to_region` (*region_name*, ***kw_params*)

Given a valid region name, return a `boto.rds2.layer1.RDSConnection`. Any additional parameters after the *region_name* are passed on to the connect method of the region object.

Type `str`

Parameters *region_name* – The name of the region to connect to.

Return type `boto.rds2.layer1.RDSConnection` or `None`

Returns A connection to the given region, or `None` if an invalid region name is given

`boto.rds2.regions` ()

Get all available regions for the RDS service.

Return type `list`

Returns A list of `boto.regioninfo.RegionInfo`

4.143.2 boto.rds2.exceptions

exception `boto.rds2.exceptions.AuthorizationAlreadyExists` (*status*, *reason*,
body=None, **args*)

exception `boto.rds2.exceptions.AuthorizationNotFound` (*status*, *reason*, *body=None*,
**args*)

exception `boto.rds2.exceptions.AuthorizationQuotaExceeded` (*status*, *reason*,
body=None, **args*)

exception `boto.rds2.exceptions.DBInstanceAlreadyExists` (*status*, *reason*, *body=None*,
**args*)

exception `boto.rds2.exceptions.DBInstanceNotFound` (*status*, *reason*, *body=None*, **args*)

exception `boto.rds2.exceptions.DBParameterGroupAlreadyExists` (*status*, *reason*,
body=None,
**args*)

exception `boto.rds2.exceptions.DBParameterGroupNotFound` (*status*, *reason*,
body=None, **args*)

exception `boto.rds2.exceptions.DBParameterGroupQuotaExceeded` (*status*, *reason*,
body=None,
**args*)

exception `boto.rds2.exceptions.DBSecurityGroupAlreadyExists` (*status*, *reason*,
body=None, **args*)

exception `boto.rds2.exceptions.DBSecurityGroupNotFound` (*status*, *reason*, *body=None*,
**args*)

```
exception boto.rds2.exceptions.DBSecurityGroupNotSupported (status, reason,
                                                            body=None, *args)
exception boto.rds2.exceptions.DBSecurityGroupQuotaExceeded (status, reason,
                                                             body=None, *args)
exception boto.rds2.exceptions.DBSnapshotAlreadyExists (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.DBSnapshotNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.DBSubnetGroupAlreadyExists (status, reason,
                                                            body=None, *args)
exception boto.rds2.exceptions.DBSubnetGroupDoesNotCoverEnoughAZs (status,
                                                                      reason,
                                                                      body=None,
                                                                      *args)
exception boto.rds2.exceptions.DBSubnetGroupNotFound (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.DBSubnetGroupQuotaExceeded (status, reason,
                                                             body=None, *args)
exception boto.rds2.exceptions.DBSubnetQuotaExceeded (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.DBUpgradeDependencyFailure (status, reason,
                                                             body=None, *args)
exception boto.rds2.exceptions.EventSubscriptionQuotaExceeded (status, reason,
                                                                body=None,
                                                                *args)
exception boto.rds2.exceptions.InstanceQuotaExceeded (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.InsufficientDBInstanceCapacity (status, reason,
                                                                body=None,
                                                                *args)
exception boto.rds2.exceptions.InvalidDBInstanceState (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.InvalidDBParameterGroupState (status, reason,
                                                                body=None, *args)
exception boto.rds2.exceptions.InvalidDBSecurityGroupState (status, reason,
                                                                body=None, *args)
exception boto.rds2.exceptions.InvalidDBSnapshotState (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.InvalidDBSubnetGroupState (status, reason,
                                                             body=None, *args)
exception boto.rds2.exceptions.InvalidDBSubnetState (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.InvalidEventSubscriptionState (status, reason,
                                                                body=None,
                                                                *args)
exception boto.rds2.exceptions.InvalidOptionGroupState (status, reason, body=None,
                                                         *args)
exception boto.rds2.exceptions.InvalidRestore (status, reason, body=None, *args)
```

```
exception boto.rds2.exceptions.InvalidSubnet (status, reason, body=None, *args)
exception boto.rds2.exceptions.InvalidVPCNetworkState (status, reason, body=None, *args)
exception boto.rds2.exceptions.OptionGroupAlreadyExists (status, reason, body=None, *args)
exception boto.rds2.exceptions.OptionGroupNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.OptionGroupQuotaExceeded (status, reason, body=None, *args)
exception boto.rds2.exceptions.PointInTimeRestoreNotEnabled (status, reason, body=None, *args)
exception boto.rds2.exceptions.ProvisionedIopsNotAvailableInAZ (status, reason, body=None, *args)
exception boto.rds2.exceptions.ReservedDBInstanceAlreadyExists (status, reason, body=None, *args)
exception boto.rds2.exceptions.ReservedDBInstanceNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.ReservedDBInstanceQuotaExceeded (status, reason, body=None, *args)
exception boto.rds2.exceptions.ReservedDBInstancesOfferingNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.SNSInvalidTopic (status, reason, body=None, *args)
exception boto.rds2.exceptions.SNSNoAuthorization (status, reason, body=None, *args)
exception boto.rds2.exceptions.SNSTopicArnNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.SnapshotQuotaExceeded (status, reason, body=None, *args)
exception boto.rds2.exceptions.SourceNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.StorageQuotaExceeded (status, reason, body=None, *args)
exception boto.rds2.exceptions.SubnetAlreadyInUse (status, reason, body=None, *args)
exception boto.rds2.exceptions.SubscriptionAlreadyExist (status, reason, body=None, *args)
exception boto.rds2.exceptions.SubscriptionCategoryNotFound (status, reason, body=None, *args)
exception boto.rds2.exceptions.SubscriptionNotFound (status, reason, body=None, *args)
```

4.143.3 boto.rds2.layer1

class `boto.rds2.layer1.RDSConnection` (**kwargs)

Amazon Relational Database Service Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks, freeing up developers to focus on what makes their applications and businesses unique.

Amazon RDS gives you access to the capabilities of a familiar MySQL or Oracle database server. This means the code, applications, and tools you already use today with your existing MySQL or Oracle databases work with Amazon RDS without modification. Amazon RDS automatically backs up your database and maintains the database software that powers your DB instance. Amazon RDS is flexible: you can scale your database instance's compute resources and storage capacity to meet your application's demand. As with all Amazon Web Services, there are no up-front investments, and you pay only for the resources you use.

This is the Amazon RDS API Reference . It contains a comprehensive description of all Amazon RDS Query APIs and data types. Note that this API is asynchronous and some actions may require polling to determine when an action has been applied. See the parameter description to determine if a change is applied immediately or on the next instance reboot or during the maintenance window. For more information on Amazon RDS concepts and usage scenarios, go to the '[Amazon RDS User Guide](#)'.

APIVersion = '2013-09-09'

DefaultRegionEndpoint = 'rds.us-east-1.amazonaws.com'

DefaultRegionName = 'us-east-1'

ResponseError

alias of `boto.exception.JSONResponseError`

add_source_identifier_to_subscription (*subscription_name*, *source_identifier*)

Adds a source identifier to an existing RDS event notification subscription.

Parameters

- **subscription_name** (*string*) – The name of the RDS event notification subscription you want to add a source identifier to.
- **source_identifier** (*string*) –

The identifier of the event source to be added. An identifier must begin with a letter and must contain only ASCII letters, digits, and hyphens; it cannot end with a hyphen or contain two consecutive hyphens.

Constraints:

- **If the source type is a DB instance, then a *DBInstanceIdentifier*** must be supplied.
- **If the source type is a DB security group, a *DBSecurityGroupName*** must be supplied.
- **If the source type is a DB parameter group, a *DBParameterGroupName*** must be supplied.
- **If the source type is a DB snapshot, a *DBSnapshotIdentifier*** must be supplied.

add_tags_to_resource (*resource_name*, *tags*)

Adds metadata tags to an Amazon RDS resource. These tags can also be used with cost allocation reporting to track cost associated with Amazon RDS resources, or used in Condition statement in IAM policy for Amazon RDS.

For an overview on tagging Amazon RDS resources, see '[Tagging Amazon RDS Resources](#)'.

Parameters

- **resource_name** (*string*) – The Amazon RDS resource the tags will be added to. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see '[Constructing an RDS Amazon Resource Name \(ARN\)](#)'.

- **tags** (*list*) – The tags to be assigned to the Amazon RDS resource. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

authorize_db_security_group_ingress (*db_security_group_name*, *cidrip=None*,
ec2_security_group_name=None,
ec2_security_group_id=None,
ec2_security_group_owner_id=None)

Enables ingress to a DBSecurityGroup using one of two forms of authorization. First, EC2 or VPC security groups can be added to the DBSecurityGroup if the application using the database is running on EC2 or VPC instances. Second, IP ranges are available if the application accessing your database is running on the Internet. Required parameters for this API are one of CIDR range, EC2SecurityGroupId for VPC, or (EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId for non-VPC). You cannot authorize ingress from an EC2 security group in one Region to an Amazon RDS DB instance in another. You cannot authorize ingress from a VPC security group in one VPC to an Amazon RDS DB instance in another. For an overview of CIDR ranges, go to the [‘Wikipedia Tutorial’](#).

Parameters

- **db_security_group_name** (*string*) – The name of the DB security group to add authorization to.
- **cidrip** (*string*) – The IP range to authorize.
- **ec2_security_group_name** (*string*) – Name of the EC2 security group to authorize. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.
- **ec2_security_group_id** (*string*) – Id of the EC2 security group to authorize. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.
- **ec2_security_group_owner_id** (*string*) – AWS Account Number of the owner of the EC2 security group specified in the *EC2SecurityGroupName* parameter. The AWS Access Key ID is not an acceptable value. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.

copy_db_snapshot (*source_db_snapshot_identifier*, *target_db_snapshot_identifier*, *tags=None*)

Copies the specified DBSnapshot. The source DBSnapshot must be in the “available” state.

Parameters **source_db_snapshot_identifier** (*string*) – The identifier for the source DB snapshot.

Constraints:

- **Must be the identifier for a valid system snapshot in the “available” state.**

Example: *rds:mydb-2012-04-02-00-01*

Parameters **target_db_snapshot_identifier** (*string*) – The identifier for the copied snapshot.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: *my-db-snapshot*

Parameters **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_instance (*db_instance_identifier*, *allocated_storage*, *db_instance_class*, *engine*, *master_username*, *master_user_password*, *db_name=None*, *db_security_groups=None*, *vpc_security_group_ids=None*, *availability_zone=None*, *db_subnet_group_name=None*, *preferred_maintenance_window=None*, *db_parameter_group_name=None*, *backup_retention_period=None*, *preferred_backup_window=None*, *port=None*, *multi_az=None*, *engine_version=None*, *auto_minor_version_upgrade=None*, *license_model=None*, *iops=None*, *option_group_name=None*, *character_set_name=None*, *publicly_accessible=None*, *tags=None*)

Creates a new DB instance.

Parameters **db_name** (*string*) – The meaning of this parameter differs according to the database engine you use.

MySQL

The name of the database to create when the DB instance is created. If this parameter is not specified, no database is created in the DB instance.

Constraints:

- Must contain 1 to 64 alphanumeric characters
- Cannot be a word reserved by the specified database engine

Type: String

Oracle

The Oracle System ID (SID) of the created DB instance.

Default: *ORCL*

Constraints:

- Cannot be longer than 8 characters

SQL Server

Not applicable. Must be null.

Parameters **db_instance_identifier** (*string*) – The DB instance identifier. This parameter is stored as a lowercase string.

Constraints:

- **Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).**
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Example: *mydbinstance*

Parameters **allocated_storage** (*integer*) – The amount of storage (in gigabytes) to be initially allocated for the database instance.

MySQL

Constraints: Must be an integer from 5 to 1024.

Type: Integer

Oracle

Constraints: Must be an integer from 10 to 1024.

SQL Server

Constraints: Must be an integer from 200 to 1024 (Standard Edition and Enterprise Edition) or from 30 to 1024 (Express Edition and Web Edition)

Parameters **db_instance_class** (*string*) – The compute and memory capacity of the DB instance.

Valid Values: `'db.t1.micro | db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.xlarge | db.m2.2xlarge | db.m2.4xlarge'`

Parameters engine (*string*) – The name of the database engine to be used for this instance.

Valid Values: *MySQL* | *oracle-se1* | *oracle-se* | *oracle-ee* | *sqlserver-ee* | *sqlserver-se* | *sqlserver-ex* | *sqlserver-web*

Parameters master_username (*string*) –

The name of master user for the client DB instance.

MySQL

Constraints:

- Must be 1 to 16 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

Type: String

Oracle

Constraints:

- Must be 1 to 30 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

SQL Server

Constraints:

- Must be 1 to 128 alphanumeric characters.
- First character must be a letter.
- Cannot be a reserved word for the chosen database engine.

Parameters master_user_password (*string*) – The password for the master database user. Can be any printable ASCII character except “/”, “”, or “@”.

Type: String

MySQL

Constraints: Must contain from 8 to 41 characters.

Oracle

Constraints: Must contain from 8 to 30 characters.

SQL Server

Constraints: Must contain from 8 to 128 characters.

Parameters db_security_groups (*list*) – A list of DB security groups to associate with this DB instance.

Default: The default DB security group for the database engine.

Parameters vpc_security_group_ids (*list*) – A list of EC2 VPC security groups to associate with this DB instance.

Default: The default EC2 VPC security group for the DB subnet group’s VPC.

Parameters availability_zone (*string*) – The EC2 Availability Zone that the database instance will be created in.

Default: A random, system-chosen Availability Zone in the endpoint’s region.

Example: *us-east-1d*

Constraint: The AvailabilityZone parameter cannot be specified if the MultiAZ parameter is set to *True*. The specified Availability Zone must be in the same region as the current endpoint.

Parameters `db_subnet_group_name` (*string*) – A DB subnet group to associate with this DB instance.

If there is no DB subnet group, then it is a non-VPC DB instance.

Parameters `preferred_maintenance_window` (*string*) – The weekly time range (in UTC) during which system maintenance can occur.

Format: *ddd:hh24:mi-ddd:hh24:mi*

Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week. To see the time blocks available, see ‘Adjusting the Preferred Maintenance Window’ in the Amazon RDS User Guide.

Valid Days: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Constraints: Minimum 30-minute window.

Parameters `db_parameter_group_name` (*string*) –

The name of the DB parameter group to associate with this DB instance. If this argument is omitted, the default DBParameterGroup for the specified engine will be used.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `backup_retention_period` (*integer*) –

The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 8
- **Cannot be set to 0 if the DB instance is a master instance with read** replicas

Parameters `preferred_backup_window` (*string*) – The daily time range during which automated backups are created if automated backups are enabled, using the *BackupRetentionPeriod* parameter.

Default: A 30-minute window selected at random from an 8-hour block of time per region. See the Amazon RDS User Guide for the time blocks for each region from which the default backup windows are assigned.

Constraints: Must be in the format *hh24:mi-hh24:mi*. Times should be Universal Time Coordinated (UTC). Must not conflict with the preferred maintenance window. Must be at least 30 minutes.

Parameters `port` (*integer*) – The port number on which the database accepts connections.

MySQL

Default: *3306*

Valid Values: *1150-65535*

Type: Integer

Oracle

Default: *1521*

Valid Values: *1150-65535*

SQL Server

Default: *1433*

Valid Values: *1150-65535* except for *1434* and *3389*.

Parameters

- **multi_az** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment. You cannot set the AvailabilityZone parameter if the MultiAZ parameter is set to true.
- **engine_version** (*string*) – The version number of the database engine to use.

MySQL

Example: *5.1.42*

Type: String

Oracle

Example: *11.2.0.2.v2*

Type: String

SQL Server

Example: *10.50.2789.0.v1*

Parameters auto_minor_version_upgrade (*boolean*) – Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window.

Default: *True*

Parameters license_model (*string*) – License model information for this DB instance.

Valid values: *license-included* | *bring-your-own-license* | **‘general-** public-license‘

Parameters iops (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.

Constraints: Must be an integer greater than 1000.

Parameters option_group_name (*string*) – Indicates that the DB instance should be associated with the specified option group.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

Parameters

- **character_set_name** (*string*) – For supported engines, indicates that the DB instance should be associated with the specified CharacterSet.
- **publicly_accessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- ****Default VPC:**** true
- ****VPC:**** false

If no DB subnet group has been specified as part of the request and the PubliclyAccessible value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has

been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be private.

Parameters `tags` (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_instance_read_replica (*db_instance_identifier*, *source_db_instance_identifier*, *db_instance_class=None*, *availability_zone=None*, *port=None*, *auto_minor_version_upgrade=None*, *iops=None*, *option_group_name=None*, *publicly_accessible=None*, *tags=None*)

Creates a DB instance that acts as a read replica of a source DB instance.

All read replica DB instances are created as Single-AZ deployments with backups disabled. All other DB instance attributes (including DB security groups and DB parameter groups) are inherited from the source DB instance, except as specified below.

The source DB instance must have backup retention enabled.

Parameters

- **db_instance_identifier** (*string*) – The DB instance identifier of the read replica. This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string.
- **source_db_instance_identifier** (*string*) – The identifier of the DB instance that will act as the source for the read replica. Each DB instance can have up to five read replicas.

Constraints: Must be the identifier of an existing DB instance that is not already a read replica DB instance.

Parameters `db_instance_class` (*string*) – The compute and memory capacity of the read replica.

Valid Values: ‘db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge

db.m2.xlarge | db.m2.2xlarge | db.m2.4xlarge’

Default: Inherits from the source DB instance.

Parameters `availability_zone` (*string*) – The Amazon EC2 Availability Zone that the read replica will be created in.

Default: A random, system-chosen Availability Zone in the endpoint’s region.

Example: *us-east-1d*

Parameters `port` (*integer*) – The port number that the DB instance uses for connections.

Default: Inherits from the source DB instance

Valid Values: *1150-65535*

Parameters `auto_minor_version_upgrade` (*boolean*) – Indicates that minor engine upgrades will be applied automatically to the read replica during the maintenance window.

Default: Inherits from the source DB instance

Parameters

- **iops** (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.
- **option_group_name** (*string*) – The option group the DB instance will be associated with. If omitted, the default option group for the engine specified will be used.

- **publicly_accessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- ****Default VPC:****true
- ****VPC:****false

If no DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be private.

Parameters `tags` (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_parameter_group (*db_parameter_group_name*, *db_parameter_group_family*, *description*, *tags=None*)

Creates a new DB parameter group.

A DB parameter group is initially created with the default parameters for the database engine used by the DB instance. To provide custom values for any of the parameters, you must modify the group after creating it using `ModifyDBParameterGroup`. Once you’ve created a DB parameter group, you need to associate it with your DB instance using `ModifyDBInstance`. When you associate a new DB parameter group with a running DB instance, you need to reboot the DB Instance for the new DB parameter group and associated settings to take effect.

Parameters `db_parameter_group_name` (*string*) –

The name of the DB parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

This value is stored as a lower-case string.

Parameters

- **db_parameter_group_family** (*string*) – The DB parameter group family name. A DB parameter group can be associated with one and only one DB parameter group family, and can be applied only to a DB instance running a database engine and engine version compatible with that DB parameter group family.
- **description** (*string*) – The description for the DB parameter group.
- **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_security_group (*db_security_group_name*, *db_security_group_description*, *tags=None*)

Creates a new DB security group. DB security groups control access to a DB instance.

Parameters `db_security_group_name` (*string*) – The name for the DB security group. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

- Must not be “Default”
- May not contain spaces

Example: *mysecuritygroup*

Parameters

- **db_security_group_description** (*string*) – The description for the DB security group.
- **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_snapshot (*db_snapshot_identifier*, *db_instance_identifier*, *tags=None*)

Creates a DBSnapshot. The source DBInstance must be in “available” state.

Parameters **db_snapshot_identifier** (*string*) – The identifier for the DB snapshot.

Constraints:

- Cannot be null, empty, or blank
- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: *my-snapshot-id*

Parameters **db_instance_identifier** (*string*) –

The DB instance identifier. This is the unique key that identifies a DB instance. This parameter isn’t case sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_db_subnet_group (*db_subnet_group_name*, *db_subnet_group_description*, *subnet_ids*, *tags=None*)

Creates a new DB subnet group. DB subnet groups must contain at least one subnet in at least two AZs in the region.

Parameters **db_subnet_group_name** (*string*) – The name for the DB subnet group. This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens. Must not be “Default”.

Example: *mySubnetgroup*

Parameters

- **db_subnet_group_description** (*string*) – The description for the DB subnet group.
- **subnet_ids** (*list*) – The EC2 Subnet IDs for the DB subnet group.
- **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

create_event_subscription (*subscription_name*, *sns_topic_arn*, *source_type=None*, *event_categories=None*, *source_ids=None*, *enabled=None*, *tags=None*)

Creates an RDS event notification subscription. This action requires a topic ARN (Amazon Resource Name) created by either the RDS console, the SNS console, or the SNS API. To obtain an ARN with

SNS, you must create a topic in Amazon SNS and subscribe to the topic. The ARN is displayed in the SNS console.

You can specify the type of source (`SourceType`) you want to be notified of, provide a list of RDS sources (`SourceIds`) that triggers the events, and provide a list of event categories (`EventCategories`) for events you want to be notified of. For example, you can specify `SourceType = db-instance`, `SourceIds = mydbinstance1, mydbinstance2` and `EventCategories = Availability, Backup`.

If you specify both the `SourceType` and `SourceIds`, such as `SourceType = db-instance` and `SourceIdentifier = myDBInstance1`, you will be notified of all the `db-instance` events for the specified source. If you specify a `SourceType` but do not specify a `SourceIdentifier`, you will receive notice of the events for that source type for all your RDS sources. If you do not specify either the `SourceType` nor the `SourceIdentifier`, you will be notified of events generated from all RDS sources belonging to your customer account.

Parameters `subscription_name` (*string*) – The name of the subscription.

Constraints: The name must be less than 255 characters.

Parameters

- `sns_topic_arn` (*string*) – The Amazon Resource Name (ARN) of the SNS topic created for event notification. The ARN is created by Amazon SNS when you create a topic and subscribe to it.
- `source_type` (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a DB instance, you would set this parameter to `db-instance`. If this value is not specified, all events are returned.

Valid values: `db-instance` | `db-parameter-group` | `db-security-group` | `db-snapshot`

Parameters

- `event_categories` (*list*) – A list of event categories for a `SourceType` that you want to subscribe to. You can see a list of the categories for a given `SourceType` in the **‘Events’** topic in the Amazon RDS User Guide or by using the `DescribeEventCategories` action.
- `source_ids` (*list*) –

The list of identifiers of the event sources for which events will be returned. If not specified, then all sources are included in the response. An identifier must begin with a letter and must contain only ASCII letters, digits, and hyphens; it cannot end with a hyphen or contain two consecutive hyphens.

Constraints:

- If `SourceIds` are supplied, `SourceType` must also be provided.
- **If the source type is a DB instance, then a `DBInstanceIdentifier` must be supplied.**
- **If the source type is a DB security group, a `DBSecurityGroupName` must be supplied.**
- **If the source type is a DB parameter group, a `DBParameterGroupName` must be supplied.**
- **If the source type is a DB snapshot, a `DBSnapshotIdentifier` must be supplied.**

Parameters

- `enabled` (*boolean*) – A Boolean value; set to **true** to activate the subscription, set to **false** to create the subscription but not active it.
- `tags` (*list*) – A list of tags. Tags must be passed as tuples in the form `[('key1', 'valueForKey1'), ('key2', 'valueForKey2')]`

`create_option_group` (*option_group_name, engine_name, major_engine_version, option_group_description, tags=None*)

Creates a new option group. You can create up to 20 option groups.

Parameters `option_group_name` (*string*) – Specifies the name of the option group to be created.

Constraints:

- Must be 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: *myoptiongroup*

Parameters

- **engine_name** (*string*) – Specifies the name of the engine that this option group should be associated with.
- **major_engine_version** (*string*) – Specifies the major version of the engine that this option group should be associated with.
- **option_group_description** (*string*) – The description of the option group.
- **tags** (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

delete_db_instance (*db_instance_identifier*, *skip_final_snapshot=None*, *final_db_snapshot_identifier=None*)

The DeleteDBInstance action deletes a previously provisioned DB instance. A successful response from the web service indicates the request was received correctly. When you delete a DB instance, all automated backups for that instance are deleted and cannot be recovered. Manual DB snapshots of the DB instance to be deleted are not deleted.

If a final DB snapshot is requested the status of the RDS instance will be “deleting” until the DB snapshot is created. The API action *DescribeDBInstance* is used to monitor the status of this operation. The action cannot be canceled or reverted once submitted.

Parameters **db_instance_identifier** (*string*) –

The DB instance identifier for the DB instance to be deleted. This parameter isn’t case sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **skip_final_snapshot** (*boolean*) – Determines whether a final DB snapshot is created before the DB instance is deleted. If *True* is specified, no DBSnapshot is created. If *false* is specified, a DB snapshot is created before the DB instance is deleted.

The FinalDBSnapshotIdentifier parameter must be specified if SkipFinalSnapshot is *False*.

Default: *False*

Parameters **final_db_snapshot_identifier** (*string*) –

The DBSnapshotIdentifier of the new DBSnapshot created when SkipFinalSnapshot is set to *False*. **Specifying this parameter and also setting the SkipFinalShapshot** parameter to true results in an error.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

delete_db_parameter_group (*db_parameter_group_name*)

Deletes a specified DBParameterGroup. The DBParameterGroup cannot be associated with any RDS instances to be deleted. The specified DB parameter group cannot be associated with any DB instances.

Parameters **db_parameter_group_name** (*string*) –

The name of the DB parameter group.

Constraints:

- Must be the name of an existing DB parameter group
- You cannot delete a default DB parameter group
- Cannot be associated with any DB instances

delete_db_security_group (*db_security_group_name*)

Deletes a DB security group. The specified DB security group must not be associated with any DB instances.

Parameters **db_security_group_name** (*string*) –

The name of the DB security group to delete.

You cannot delete the default DB security group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- Must not be “Default”
- May not contain spaces

delete_db_snapshot (*db_snapshot_identifier*)

Deletes a DBSnapshot. The DBSnapshot must be in the *available* state to be deleted.

Parameters **db_snapshot_identifier** (*string*) – The DBSnapshot identifier.

Constraints: Must be the name of an existing DB snapshot in the *available* state.

delete_db_subnet_group (*db_subnet_group_name*)

Deletes a DB subnet group. The specified database subnet group must not be associated with any DB instances.

Parameters **db_subnet_group_name** (*string*) –

The name of the database subnet group to delete.

You cannot delete the default subnet group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

delete_event_subscription (*subscription_name*)

Deletes an RDS event notification subscription.

Parameters **subscription_name** (*string*) – The name of the RDS event notification subscription you want to delete.

delete_option_group (*option_group_name*)

Deletes an existing option group.

Parameters **option_group_name** (*string*) –

The name of the option group to be deleted.

You cannot delete default option groups.

describe_db_engine_versions (*engine=None,* *engine_version=None,*
db_parameter_group_family=None, *max_records=None,*
marker=None, *default_only=None,*
list_supported_character_sets=None)

Returns a list of the available DB engines.

Parameters

- **engine** (*string*) – The database engine to return.

- **engine_version** (*string*) – The database engine version to return.

Example: 5.1.49

Parameters **db_parameter_group_family** (*string*) –

The name of a specific DB parameter group family to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **max_records** (*integer*) – The maximum number of records to include in the response. If more than the *MaxRecords* value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters

- **marker** (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.
- **default_only** (*boolean*) – Indicates that only the default version of the specified engine or engine and major version combination is returned.
- **list_supported_character_sets** (*boolean*) – If this parameter is specified, and if the requested engine supports the *CharacterSetName* parameter for *CreateDBInstance*, the response includes a list of supported character sets for each engine version.

describe_db_instances (*db_instance_identifier=None, filters=None, max_records=None, marker=None*)

Returns information about provisioned RDS instances. This API supports pagination.

Parameters **db_instance_identifier** (*string*) –

The user-supplied instance identifier. If this parameter is specified, information from only the specific DB instance is returned. This parameter isn't case sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters

- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters **marker** (*string*) – An optional pagination token provided by a previous *DescribeDBInstances* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_db_log_files (*db_instance_idenfifier*, *filename_contains=None*,
file_last_written=None, *file_size=None*, *max_records=None*,
marker=None)

Returns a list of DB log files for the DB instance.

Parameters **db_instance_idenfifier** (*string*) –

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters

- **filename_contains** (*string*) – Filters the available log files for log file names that contain the specified string.
- **file_last_written** (*long*) – Filters the available log files for files written since the specified date, in POSIX timestamp format.
- **file_size** (*long*) – Filters the available log files for files larger than the specified size.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified MaxRecords value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.
- **marker** (*string*) – The pagination token provided in the previous request. If this parameter is specified the response includes only records beyond the marker, up to MaxRecords.

describe_db_parameter_groups (*db_parameter_group_name=None*, *filters=None*,
max_records=None, *marker=None*)

Returns a list of *DBParameterGroup* descriptions. If a *DBParameterGroupName* is specified, the list will contain only the description of the specified DB parameter group.

Parameters **db_parameter_group_name** (*string*) –

The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters

- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters **marker** (*string*) – An optional pagination token provided by a previous *DescribeDBParameterGroups* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_db_parameters (*db_parameter_group_name*, *source=None*, *max_records=None*, *marker=None*)

Returns the detailed parameter list for a particular DB parameter group.

Parameters *db_parameter_group_name* (*string*) –

The name of a specific DB parameter group to return details for.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters *source* (*string*) – The parameter types to return.

Default: All parameter types returned

Valid Values: *user* | *system* | *engine-default*

Parameters *max_records* (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters *marker* (*string*) – An optional pagination token provided by a previous *DescribeDBParameters* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_db_security_groups (*db_security_group_name=None*, *filters=None*, *max_records=None*, *marker=None*)

Returns a list of *DBSecurityGroup* descriptions. If a *DBSecurityGroupName* is specified, the list will contain only the descriptions of the specified DB security group.

Parameters

- **db_security_group_name** (*string*) – The name of the DB security group to return details for.
- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters *marker* (*string*) – An optional pagination token provided by a previous *DescribeDBSecurityGroups* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_db_snapshots (*db_instance_identifier=None*, *db_snapshot_identifier=None*, *snapshot_type=None*, *filters=None*, *max_records=None*, *marker=None*)

Returns information about DB snapshots. This API supports pagination.

Parameters *db_instance_identifier* (*string*) –

A DB instance identifier to retrieve the list of DB snapshots for. Cannot be used in conjunction with *DBSnapshotIdentifier*. This parameter is not case sensitive.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `db_snapshot_identifier` (*string*) –

A specific DB snapshot identifier to describe. Cannot be used in conjunction with *DBInstanceIdentifier*. This value is stored as a lowercase string.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens
- **If this is the identifier of an automated snapshot, the *SnapshotType* parameter must also be specified.**

Parameters

- **snapshot_type** (*string*) – The type of snapshots that will be returned. Values can be “automated” or “manual.” If not specified, the returned results will include all snapshots types.
- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters `marker` (*string*) – An optional pagination token provided by a previous *DescribeDBSnapshots* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_db_subnet_groups (*db_subnet_group_name=None, filters=None, max_records=None, marker=None*)

Returns a list of DBSubnetGroup descriptions. If a DBSubnetGroupName is specified, the list will contain only the descriptions of the specified DBSubnetGroup.

For an overview of CIDR ranges, go to the [‘Wikipedia Tutorial’](#).

Parameters

- **db_subnet_group_name** (*string*) – The name of the DB subnet group to return details for.
- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters `marker` (*string*) – An optional pagination token provided by a previous *DescribeDBSubnetGroups* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_engine_default_parameters (*db_parameter_group_family, max_records=None, marker=None*)

Returns the default engine and system parameter information for the specified database engine.

Parameters

- **db_parameter_group_family** (*string*) – The name of the DB parameter group family.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous *DescribeEngineDefaultParameters* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_event_categories (*source_type=None*)

Displays a list of categories for all event source types, or, if specified, for a specified source type. You can see a list of the event categories and source types in the ‘Events’_ topic in the Amazon RDS User Guide.

Parameters source_type (*string*) – The type of source that will be generating the events.

Valid values: db-instance | db-parameter-group | db-security-group | db-snapshot

describe_event_subscriptions (*subscription_name=None, filters=None, max_records=None, marker=None*)

Lists all the subscription descriptions for a customer account. The description for a subscription includes SubscriptionName, SNSTopicARN, CustomerID, SourceType, SourceID, CreationTime, and Status.

If you specify a SubscriptionName, lists the description for that subscription.

Parameters

- **subscription_name** (*string*) – The name of the RDS event notification subscription you want to describe.
- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous *DescribeOrderableDBInstanceOptions* request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_events (*source_identifier=None, source_type=None, start_time=None, end_time=None, duration=None, event_categories=None, max_records=None, marker=None*)

Returns events related to DB instances, DB security groups, DB snapshots, and DB parameter groups for the past 14 days. Events specific to a particular DB instance, DB security group, database snapshot, or DB parameter group can be obtained by providing the name as a parameter. By default, the past hour of events are returned.

Parameters source_identifier (*string*) –

The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.

Constraints:

- If *SourceIdentifier* is supplied, *SourceType* must also be provided.
- **If the source type is *DBInstance*, then a *DBInstanceIdentifier* must be supplied.**
- **If the source type is *DBSecurityGroup*, a *DBSecurityGroupName* must be supplied.**

- If the source type is *DBParameterGroup*, a *DBParameterGroupName* must be supplied.
- If the source type is *DBSnapshot*, a *DBSnapshotIdentifier* must be supplied.
- Cannot end with a hyphen or contain two consecutive hyphens.

Parameters

- **source_type** (*string*) – The event source to retrieve events for. If no value is specified, all events are returned.
- **start_time** (*timestamp*) – The beginning of the time interval to retrieve events for, specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#)

Example: 2009-07-08T18:00Z

Parameters end_time (*timestamp*) – The end of the time interval for which to retrieve events, specified in ISO 8601 format. For more information about ISO 8601, go to the [‘ISO8601 Wikipedia page.’](#)

Example: 2009-07-08T18:00Z

Parameters duration (*integer*) – The number of minutes to retrieve events for.

Default: 60

Parameters

- **event_categories** (*list*) – A list of event categories that trigger notifications for a event notification subscription.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results may be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous `DescribeEvents` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_option_group_options (*engine_name*, *major_engine_version=None*,
max_records=None, *marker=None*)

Describes all available options.

Parameters

- **engine_name** (*string*) – A required parameter. Options available for the given Engine name will be described.
- **major_engine_version** (*string*) – If specified, filters the results to include only options for the specified major engine version.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified *MaxRecords* value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_option_groups (*option_group_name=None*, *filters=None*, *marker=None*,
max_records=None, *engine_name=None*, *major_engine_version=None*)

Describes the available option groups.

Parameters

- **option_group_name** (*string*) – The name of the option group to describe. Cannot be supplied together with `EngineName` or `MajorEngineVersion`.
- **filters** (*list*) –
- **marker** (*string*) – An optional pagination token provided by a previous `DescribeOptionGroups` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters

- **engine_name** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine.
- **major_engine_version** (*string*) – Filters the list of option groups to only include groups associated with a specific database engine version. If specified, then `EngineName` must also be specified.

describe_orderable_db_instance_options (*engine*, *engine_version=None*,
db_instance_class=None, *license_model=None*, *vpc=None*,
max_records=None, *marker=None*)

Returns a list of orderable DB instance options for the specified engine.

Parameters

- **engine** (*string*) – The name of the engine to retrieve DB instance options for.
- **engine_version** (*string*) – The engine version filter value. Specify this parameter to show only the available offerings matching the specified engine version.
- **db_instance_class** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **license_model** (*string*) – The license model filter value. Specify this parameter to show only the available offerings matching the specified license model.
- **vpc** (*boolean*) – The VPC filter value. Specify this parameter to show only the available VPC or non-VPC offerings.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more records exist than the specified `MaxRecords` value, a pagination token called a marker is included in the response so that the remaining results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters **marker** (*string*) – An optional pagination token provided by a previous `DescribeOrderableDBInstanceOptions` request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by `MaxRecords`.

describe_reserved_db_instances (*reserved_db_instance_id=None*, *reserved_db_instances_offering_id=None*, *db_instance_class=None*, *duration=None*, *product_description=None*, *offering_type=None*, *multi_az=None*, *filters=None*, *max_records=None*, *marker=None*)

Returns information about reserved DB instances for this account, or about a specified reserved DB instance.

Parameters

- **reserved_db_instance_id** (*string*) – The reserved DB instance identifier filter value. Specify this parameter to show only the reservation that matches the specified reservation ID.
- **reserved_db_instances_offering_id** (*string*) – The offering identifier filter value. Specify this parameter to show only purchased reservations matching the specified offering identifier.
- **db_instance_class** (*string*) – The DB instance class filter value. Specify this parameter to show only those reservations matching the specified DB instances class.
- **duration** (*string*) – The duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: *1* | *3* | *31536000* | *94608000*

Parameters

- **product_description** (*string*) – The product description filter value. Specify this parameter to show only those reservations matching the specified product description.
- **offering_type** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: **“Light Utilization”** | **“Medium Utilization”** | **“Heavy Utilization”**

Parameters

- **multi_az** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only those reservations matching the specified Multi-AZ parameter.
- **filters** (*list*) –
- **max_records** (*integer*) – The maximum number of records to include in the response. If more than the *MaxRecords* value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

describe_reserved_db_instances_offerings (*reserved_db_instances_offering_id=None*, *db_instance_class=None*, *duration=None*, *product_description=None*, *offering_type=None*, *multi_az=None*, *max_records=None*, *marker=None*)

Lists available reserved DB instance offerings.

Parameters reserved_db_instances_offering_id (*string*) – The offering identifier filter value. Specify this parameter to show only the available offering that matches the specified reservation identifier.

Example: `438012d3-4052-4cc7-b2e3-8d3372e0e706`

Parameters

- **db_instance_class** (*string*) – The DB instance class filter value. Specify this parameter to show only the available offerings matching the specified DB instance class.
- **duration** (*string*) – Duration filter value, specified in years or seconds. Specify this parameter to show only reservations for this duration.

Valid Values: `1 | 3 | 31536000 | 94608000`

Parameters

- **product_description** (*string*) – Product description filter value. Specify this parameter to show only the available offerings matching the specified product description.
- **offering_type** (*string*) – The offering type filter value. Specify this parameter to show only the available offerings matching the specified offering type.

Valid Values: `“Light Utilization” | “Medium Utilization” | “Heavy Utilization”`

Parameters

- **multi_az** (*boolean*) – The Multi-AZ filter value. Specify this parameter to show only the available offerings matching the specified Multi-AZ parameter.
- **max_records** (*integer*) – The maximum number of records to include in the response. If more than the *MaxRecords* value is available, a pagination token called a marker is included in the response so that the following results can be retrieved.

Default: 100

Constraints: minimum 20, maximum 100

Parameters marker (*string*) – An optional pagination token provided by a previous request. If this parameter is specified, the response includes only records beyond the marker, up to the value specified by *MaxRecords*.

download_db_log_file_portion (*db_instance_identifier*, *log_file_name*, *marker=None*, *number_of_lines=None*)

Downloads the last line of the specified log file.

Parameters db_instance_identifier (*string*) –

The customer-assigned name of the DB instance that contains the log files you want to list.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters

- **log_file_name** (*string*) – The name of the log file to be downloaded.
- **marker** (*string*) – The pagination token provided in the previous request. If this parameter is specified the response includes only records beyond the marker, up to *MaxRecords*.
- **number_of_lines** (*integer*) – The number of lines remaining to be downloaded.

list_tags_for_resource (*resource_name*)

Lists all tags on an Amazon RDS resource.

For an overview on tagging an Amazon RDS resource, see ‘[Tagging Amazon RDS Resources](#)’.

Parameters **resource_name** (*string*) – The Amazon RDS resource with tags to be listed. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see ‘[Constructing an RDS Amazon Resource Name \(ARN\)](#)’.

modify_db_instance (*db_instance_identifier*, *allocated_storage=None*,
db_instance_class=None, *db_security_groups=None*,
vpc_security_group_ids=None, *apply_immediately=None*, *master_user_password=None*, *db_parameter_group_name=None*,
backup_retention_period=None, *preferred_backup_window=None*,
preferred_maintenance_window=None, *multi_az=None*, *engine_version=None*, *allow_major_version_upgrade=None*,
auto_minor_version_upgrade=None, *iops=None*, *option_group_name=None*, *new_db_instance_identifier=None*)

Modify settings for a DB instance. You can change one or more database configuration parameters by specifying these parameters and the new values in the request.

Parameters **db_instance_identifier** (*string*) –

The DB instance identifier. This value is stored as a lowercase string.

Constraints:

- Must be the identifier for an existing DB instance
- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **allocated_storage** (*integer*) – The new storage capacity of the RDS instance. Changing this parameter does not result in an outage and the change is applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request.

MySQL

Default: Uses existing setting

Valid Values: 5-1024

Constraints: **Value supplied must be at least 10% greater than the** current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

Type: Integer

Oracle

Default: Uses existing setting

Valid Values: 10-1024

Constraints: **Value supplied must be at least 10% greater than the** current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

SQL Server

Cannot be modified.

If you choose to migrate your DB instance from using standard storage to using Provisioned IOPS, or from using Provisioned IOPS to using standard storage, the process can take time. The duration of the migration depends on several factors such as database load, storage size, storage type (standard or Provisioned IOPS), amount of IOPS provisioned (if any), and the number of prior scale storage operations. Typical migration times are under 24 hours, but the process can take up to several days in some cases. During the migration, the DB instance will be available for use, but

may experience performance degradation. While the migration takes place, nightly backups for the instance will be suspended. No other Amazon RDS operations can take place for the instance, including modifying the instance, rebooting the instance, deleting the instance, creating a read replica for the instance, and creating a DB snapshot of the instance.

Parameters `db_instance_class` (*string*) – The new compute and memory capacity of the DB instance. To determine the instance classes that are available for a particular DB engine, use the `DescribeOrderableDBInstanceOptions` action.

Passing a value for this parameter causes an outage during the change and is applied during the next maintenance window, unless the `ApplyImmediately` parameter is specified as `True` for this request.

Default: Uses existing setting

Valid Values: `'db.t1.micro | db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.xlarge | db.m2.2xlarge | db.m2.4xlarge'`

Parameters `db_security_groups` (*list*) –

A list of DB security groups to authorize on this DB instance. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `vpc_security_group_ids` (*list*) –

A list of EC2 VPC security groups to authorize on this DB instance. This change is asynchronously applied as soon as possible.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `apply_immediately` (*boolean*) – Specifies whether or not the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB instance.

If this parameter is passed as `False`, changes to the DB instance are applied on the next call to `RebootDBInstance`, the next maintenance reboot, or the next failure reboot, whichever occurs first. See each parameter to determine when a change is applied.

Default: `False`

Parameters `master_user_password` (*string*) –

The new password for the DB instance master user. Can be any printable ASCII character except `"'”, “””, or “@”`.

Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible. Between the time of the request and the completion of the request, the `MasterUserPassword` element exists in the `PendingModifiedValues` element of the operation response.

Default: Uses existing setting

Constraints: Must be 8 to 41 alphanumeric characters (MySQL), 8 to 30 alphanumeric characters (Oracle), or 8 to 128 alphanumeric characters (SQL Server).

Amazon RDS API actions never return the password, so this action provides a way to regain access to a master instance user if the password is lost.

Parameters `db_parameter_group_name` (*string*) – The name of the DB parameter group to apply to this DB instance. Changing this parameter does not result in an outage and the change is applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request.

Default: Uses existing setting

Constraints: The DB parameter group must be in the same DB parameter group family as this DB instance.

Parameters `backup_retention_period` (*integer*) –

The number of days to retain automated backups. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Changing this parameter can result in an outage if you change from 0 to a non-zero value or from a non-zero value to 0. These changes are applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request. If you change the parameter from one non-zero value to another non-zero value, the change is asynchronously applied as soon as possible.

Default: Uses existing setting

Constraints:

- Must be a value from 0 to 8
- **Cannot be set to 0 if the DB instance is a master instance with read** replicas or if the DB instance is a read replica

Parameters `preferred_backup_window` (*string*) –

The daily time range during which automated backups are created if automated backups are enabled, as determined by the *BackupRetentionPeriod*. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints:

- Must be in the format hh24:mi-hh24:mi
- Times should be Universal Time Coordinated (UTC)
- Must not conflict with the preferred maintenance window
- Must be at least 30 minutes

Parameters `preferred_maintenance_window` (*string*) – The weekly time range (in UTC) during which system maintenance can occur, which may result in an outage. Changing this parameter does not result in an outage, except in the following situation, and the change is asynchronously applied as soon as possible. If there are pending actions that cause a reboot, and the maintenance window is changed to include the current time, then changing this parameter will cause a reboot of the DB instance. If moving this window to the current time, there must be at least 30 minutes between the current time and end of the window to ensure pending changes are applied.

Default: Uses existing setting

Format: ddd:hh24:mi-ddd:hh24:mi

Valid Days: Mon | Tue | Wed | Thu | Fri | Sat | Sun

Constraints: Must be at least 30 minutes

Parameters `multi_az` (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment. Changing this parameter does not result in an outage and the change is applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request.

Constraints: Cannot be specified if the DB instance is a read replica.

Parameters `engine_version` (*string*) – The version number of the database engine to upgrade to. Changing this parameter results in an outage and the change is applied

during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request.

For major version upgrades, if a non-default DB parameter group is currently in use, a new DB parameter group in the DB parameter group family for the new engine version must be specified. The new DB parameter group can be the default for that DB parameter group family.

Example: 5.1.42

Parameters `allow_major_version_upgrade` (*boolean*) – Indicates that major version upgrades are allowed. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.

Constraints: This parameter must be set to true when specifying a value for the EngineVersion parameter that is a different major version than the DB instance's current version.

Parameters

- **`auto_minor_version_upgrade`** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window. Changing this parameter does not result in an outage except in the following case and the change is asynchronously applied as soon as possible. An outage will result if this parameter is set to *True* during the maintenance window, and a newer minor version is available, and RDS has enabled auto patching for that engine version.
- **`iops`** (*integer*) – The new Provisioned IOPS (I/O operations per second) value for the RDS instance. Changing this parameter does not result in an outage and the change is applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request.

Default: Uses existing setting

Constraints: Value supplied must be at least 10% greater than the current value. Values that are not at least 10% greater than the existing value are rounded up so that they are 10% greater than the current value.

Type: Integer

If you choose to migrate your DB instance from using standard storage to using Provisioned IOPS, or from using Provisioned IOPS to using standard storage, the process can take time. The duration of the migration depends on several factors such as database load, storage size, storage type (standard or Provisioned IOPS), amount of IOPS provisioned (if any), and the number of prior scale storage operations. Typical migration times are under 24 hours, but the process can take up to several days in some cases. During the migration, the DB instance will be available for use, but may experience performance degradation. While the migration takes place, nightly backups for the instance will be suspended. No other Amazon RDS operations can take place for the instance, including modifying the instance, rebooting the instance, deleting the instance, creating a read replica for the instance, and creating a DB snapshot of the instance.

Parameters `option_group_name` (*string*) – Indicates that the DB instance should be associated with the specified option group. Changing this parameter does not result in an outage except in the following case and the change is applied during the next maintenance window unless the *ApplyImmediately* parameter is set to *True* for this request. If the parameter change results in an option group that enables OEM, this change can cause a brief (sub-second) period during which new connections are rejected but existing connections are not interrupted.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

Parameters `new_db_instance_identifier` (*string*) –

The new DB instance identifier for the DB instance when renaming a DB Instance. This value is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

modify_db_parameter_group (*db_parameter_group_name*, *parameters*)

Modifies the parameters of a DB parameter group. To modify more than one parameter, submit a list of the following: *ParameterName*, *ParameterValue*, and *ApplyMethod*. A maximum of 20 parameters can be modified in a single request.

The *apply-immediate* method can be used only for dynamic parameters; the *pending-reboot* method can be used with MySQL and Oracle DB instances for either dynamic or static parameters. For Microsoft SQL Server DB instances, the *pending-reboot* method can be used only for static parameters.

Parameters *db_parameter_group_name* (*string*) –

The name of the DB parameter group.

Constraints:

- Must be the name of an existing DB parameter group
- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters *parameters* (*list*) –

An array of parameter names, values, and the apply method for the parameter update. At least one parameter name, value, and apply method must be supplied; subsequent arguments are optional. A maximum of 20 parameters may be modified in a single request.

Valid Values (for the application method): *immediate* | *pending-reboot*

You can use the immediate value with dynamic parameters only. You can use the *pending-reboot* value for both dynamic and static parameters, and changes are applied when DB instance reboots.

modify_db_subnet_group (*db_subnet_group_name*, *subnet_ids*,
db_subnet_group_description=None)

Modifies an existing DB subnet group. DB subnet groups must contain at least one subnet in at least two AZs in the region.

Parameters *db_subnet_group_name* (*string*) – The name for the DB subnet group.
This value is stored as a lowercase string.

Constraints: Must contain no more than 255 alphanumeric characters or hyphens. Must not be “Default”.

Example: *mySubnetgroup*

Parameters

- **db_subnet_group_description** (*string*) – The description for the DB subnet group.
- **subnet_ids** (*list*) – The EC2 subnet IDs for the DB subnet group.

modify_event_subscription (*subscription_name*, *sns_topic_arn=None*, *source_type=None*,
event_categories=None, *enabled=None*)

Modifies an existing RDS event notification subscription. Note that you cannot modify the source identifiers using this call; to change source identifiers for a subscription, use the `AddSourceIdentifierToSubscription` and `RemoveSourceIdentifierFromSubscription` calls.

You can see a list of the event categories for a given `SourceType` in the ‘**Events**’ topic in the Amazon RDS User Guide or by using the `DescribeEventCategories` action.

Parameters

- **subscription_name** (*string*) – The name of the RDS event notification subscription.
- **sns_topic_arn** (*string*) – The Amazon Resource Name (ARN) of the SNS topic created for event notification. The ARN is created by Amazon SNS when you create a topic and subscribe to it.
- **source_type** (*string*) – The type of source that will be generating the events. For example, if you want to be notified of events generated by a DB instance, you would set this parameter to `db-instance`. If this value is not specified, all events are returned.

Valid values: `db-instance` | `db-parameter-group` | `db-security-group` | `db-snapshot`

Parameters

- **event_categories** (*list*) – A list of event categories for a SourceType that you want to subscribe to. You can see a list of the categories for a given SourceType in the ‘**Events**’ topic in the Amazon RDS User Guide or by using the **DescribeEventCategories** action.
- **enabled** (*boolean*) – A Boolean value; set to **true** to activate the subscription.

modify_option_group (*option_group_name*, *options_to_include=None*, *options_to_remove=None*, *apply_immediately=None*)

Modifies an existing option group.

Parameters **option_group_name** (*string*) – The name of the option group to be modified.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

Parameters

- **options_to_include** (*list*) – Options in this list are added to the option group or, if already present, the specified configuration is used to update the existing configuration.
- **options_to_remove** (*list*) – Options in this list are removed from the option group.
- **apply_immediately** (*boolean*) – Indicates whether the changes should be applied immediately, or during the next maintenance window for each instance associated with the option group.

promote_read_replica (*db_instance_identifier*, *backup_retention_period=None*, *preferred_backup_window=None*)

Promotes a read replica DB instance to a standalone DB instance.

Parameters **db_instance_identifier** (*string*) – The DB instance identifier. This value is stored as a lowercase string.

Constraints:

- Must be the identifier for an existing read replica DB instance
- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: `mydbinstance`

Parameters **backup_retention_period** (*integer*) –

The number of days to retain automated backups. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Default: 1

Constraints:

- Must be a value from 0 to 8

Parameters preferred_backup_window (*string*) – The daily time range during which automated backups are created if automated backups are enabled, using the *BackupRetentionPeriod* parameter.

Default: A 30-minute window selected at random from an 8-hour block of time per region. See the Amazon RDS User Guide for the time blocks for each region from which the default backup windows are assigned.

Constraints: Must be in the format *hh24:mi-hh24:mi*. Times should be Universal Time Coordinated (UTC). Must not conflict with the preferred maintenance window. Must be at least 30 minutes.

purchase_reserved_db_instances_offering (*reserved_db_instances_offering_id*,
reserved_db_instance_id=None,
db_instance_count=None, *tags=None*)

Purchases a reserved DB instance offering.

Parameters reserved_db_instances_offering_id (*string*) – The ID of the Reserved DB instance offering to purchase.

Example: 438012d3-4052-4cc7-b2e3-8d3372e0e706

Parameters reserved_db_instance_id (*string*) – Customer-specified identifier to track this reservation.

Example: myreservationID

Parameters db_instance_count (*integer*) – The number of instances to reserve.

Default: 1

Parameters tags (*list*) – A list of tags. Tags must be passed as tuples in the form [(‘key1’, ‘valueForKey1’), (‘key2’, ‘valueForKey2’)]

reboot_db_instance (*db_instance_identifier*, *force_failover=None*)

Rebooting a DB instance restarts the database engine service. A reboot also applies to the DB instance any modifications to the associated DB parameter group that were pending. Rebooting a DB instance results in a momentary outage of the instance, during which the DB instance status is set to rebooting. If the RDS instance is configured for MultiAZ, it is possible that the reboot will be conducted through a failover. An Amazon RDS event is created when the reboot is completed.

If your DB instance is deployed in multiple Availability Zones, you can force a failover from one AZ to the other during the reboot. You might force a failover to test the availability of your DB instance deployment or to restore operations to the original AZ after a failover occurs.

The time required to reboot is a function of the specific database engine’s crash recovery process. To improve the reboot time, we recommend that you reduce database activities as much as possible during the reboot process to reduce rollback activity for in-transit transactions.

Parameters db_instance_identifier (*string*) –

The DB instance identifier. This parameter is stored as a lowercase string.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters force_failover (*boolean*) – When *True*, the reboot will be conducted through a MultiAZ failover.

Constraint: You cannot specify *True* if the instance is not configured for MultiAZ.

remove_source_identifier_from_subscription (*subscription_name*, *source_identifier*)

Removes a source identifier from an existing RDS event notification subscription.

Parameters

- **subscription_name** (*string*) – The name of the RDS event notification subscription you want to remove a source identifier from.
- **source_identifier** (*string*) – The source identifier to be removed from the subscription, such as the **DB instance identifier** for a DB instance or the name of a security group.

remove_tags_from_resource (*resource_name*, *tag_keys*)

Removes metadata tags from an Amazon RDS resource.

For an overview on tagging an Amazon RDS resource, see [‘Tagging Amazon RDS Resources’](#).

Parameters

- **resource_name** (*string*) – The Amazon RDS resource the tags will be removed from. This value is an Amazon Resource Name (ARN). For information about creating an ARN, see [‘Constructing an RDS Amazon Resource Name \(ARN\)’](#).
- **tag_keys** (*list*) – The tag key (name) of the tag to be removed.

reset_db_parameter_group (*db_parameter_group_name*, *reset_all_parameters=None*, *parameters=None*)

Modifies the parameters of a DB parameter group to the engine/system default value. To reset specific parameters submit a list of the following: *ParameterName* and *ApplyMethod*. To reset the entire DB parameter group, specify the *DBParameterGroup* name and *ResetAllParameters* parameters. When resetting the entire group, dynamic parameters are updated immediately and static parameters are set to *pending-reboot* to take effect on the next DB instance restart or *RebootDBInstance* request.

Parameters **db_parameter_group_name** (*string*) –

The name of the DB parameter group.

Constraints:

- Must be 1 to 255 alphanumeric characters
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **reset_all_parameters** (*boolean*) – Specifies whether (*True*) or not (*False*) to reset all parameters in the DB parameter group to default values.

Default: *True*

Parameters **parameters** (*list*) – An array of parameter names, values, and the apply method for the parameter update. At least one parameter name, value, and apply method must be supplied; subsequent arguments are optional. A maximum of 20 parameters may be modified in a single request.

MySQL

Valid Values (for Apply method): *immediate* | *pending-reboot*

You can use the immediate value with dynamic parameters only. You can use the *pending-reboot* value for both dynamic and static parameters, and changes are applied when DB instance reboots.

Oracle

Valid Values (for Apply method): *pending-reboot*

restore_db_instance_from_db_snapshot (*db_instance_identifier*, *db_snapshot_identifier*, *db_instance_class=None*, *port=None*, *availability_zone=None*, *db_subnet_group_name=None*, *multi_az=None*, *publicly_accessible=None*, *auto_minor_version_upgrade=None*, *license_model=None*, *db_name=None*, *engine=None*, *iops=None*, *option_group_name=None*, *tags=None*)

Creates a new DB instance from a DB snapshot. The target database is created from the source database restore point with the same configuration as the original source database, except that the new RDS instance is created with the default security group.

Parameters **db_instance_identifier** (*string*) –

The identifier for the DB snapshot to restore from.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters **db_snapshot_identifier** (*string*) – Name of the DB instance to create from the DB snapshot. This parameter isn't case sensitive.

Constraints:

- Must contain from 1 to 255 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Example: *my-snapshot-id*

Parameters **db_instance_class** (*string*) – The compute and memory capacity of the Amazon RDS DB instance.

Valid Values: `'db.t1.micro | db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.2xlarge | db.m2.4xlarge'`

Parameters **port** (*integer*) – The port number on which the database accepts connections.

Default: The same port as the original DB instance

Constraints: Value must be *1150-65535*

Parameters **availability_zone** (*string*) – The EC2 Availability Zone that the database instance will be created in.

Default: A random, system-chosen Availability Zone.

Constraint: You cannot specify the **AvailabilityZone** parameter if the **MultiAZ** parameter is set to *True*.

Example: *us-east-1a*

Parameters

- **db_subnet_group_name** (*string*) – The DB subnet group name to use for the new instance.
- **multi_az** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment.

Constraint: You cannot specify the **AvailabilityZone** parameter if the **MultiAZ** parameter is set to *True*.

Parameters **publicly_accessible** (*boolean*) – Specifies the accessibility options for the DB instance. A value of *true* specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of *false* specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- ****Default VPC:****true
- ****VPC:****false

If no DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be private.

Parameters

- **auto_minor_version_upgrade** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window.
- **license_model** (*string*) – License model information for the restored DB instance.

Default: Same as source.

Valid values: `license-included` | `bring-your-own-license` | `'general-public-license'`

Parameters `db_name` (*string*) –

The database name for the restored DB instance.

This parameter doesn't apply to the MySQL engine.

Parameters `engine` (*string*) – The database engine to use for the new instance.

Default: The same as source

Constraint: Must be compatible with the engine of the source

Example: `oracle-ee`

Parameters `iops` (*integer*) – Specifies the amount of provisioned IOPS for the DB instance, expressed in I/O operations per second. If this parameter is not specified, the IOPS value will be taken from the backup. If this parameter is set to 0, the new instance will be converted to a non-PIOPS instance, which will take additional time, though your DB instance will be available for connections before the conversion starts.

Constraints: Must be an integer greater than 1000.

Parameters `option_group_name` (*string*) – The name of the option group to be used for the restored DB instance.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

Parameters `tags` (*list*) – A list of tags. Tags must be passed as tuples in the form `[('key1', 'valueForKey1'), ('key2', 'valueForKey2')]`

restore_db_instance_to_point_in_time (*source_db_instance_identifier*, *target_db_instance_identifier*, *restore_time=None*, *use_latest_restorable_time=None*, *db_instance_class=None*, *port=None*, *availability_zone=None*, *db_subnet_group_name=None*, *multi_az=None*, *publicly_accessible=None*, *auto_minor_version_upgrade=None*, *license_model=None*, *db_name=None*, *engine=None*, *iops=None*, *option_group_name=None*, *tags=None*)

Restores a DB instance to an arbitrary point-in-time. Users can restore to any point in time before the

latestRestorableTime for up to backupRetentionPeriod days. The target database is created from the source database with the same configuration as the original database except that the DB instance is created with the default DB security group.

Parameters `source_db_instance_identifier` (*string*) –

The identifier of the source DB instance from which to restore.

Constraints:

- Must be the identifier of an existing database instance
- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `target_db_instance_identifier` (*string*) –

The name of the new database instance to be created.

Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens
- First character must be a letter
- Cannot end with a hyphen or contain two consecutive hyphens

Parameters `restore_time` (*timestamp*) – The date and time to restore from.

Valid Values: Value must be a UTC time

Constraints:

- Must be before the latest restorable time for the DB instance
- Cannot be specified if UseLatestRestorableTime parameter is true

Example: `2009-09-07T23:45:00Z`

Parameters `use_latest_restorable_time` (*boolean*) – Specifies whether (*True*) or not (*False*) the DB instance is restored from the latest backup time.

Default: *False*

Constraints: Cannot be specified if RestoreTime parameter is provided.

Parameters `db_instance_class` (*string*) – The compute and memory capacity of the Amazon RDS DB instance.

Valid Values: `'db.t1.micro | db.m1.small | db.m1.medium | db.m1.large | db.m1.xlarge | db.m2.2xlarge | db.m2.4xlarge'`

Default: The same DBInstanceClass as the original DB instance.

Parameters `port` (*integer*) – The port number on which the database accepts connections.

Constraints: Value must be `1150-65535`

Default: The same port as the original DB instance.

Parameters `availability_zone` (*string*) – The EC2 Availability Zone that the database instance will be created in.

Default: A random, system-chosen Availability Zone.

Constraint: You cannot specify the `AvailabilityZone` parameter if the `MultiAZ` parameter is set to `true`.

Example: `us-east-1a`

Parameters

- **`db_subnet_group_name`** (*string*) – The DB subnet group name to use for the new instance.
- **`multi_az`** (*boolean*) – Specifies if the DB instance is a Multi-AZ deployment.

Constraint: You cannot specify the `AvailabilityZone` parameter if the `MultiAZ` parameter is set to `True`.

Parameters `publicly_accessible` (*boolean*) – Specifies the accessibility options for the DB instance. A value of `true` specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of `false` specifies an internal instance with a DNS name that resolves to a private IP address.

Default: The default behavior varies depending on whether a VPC has been requested or not. The following list shows the default behavior in each case.

- `**Default VPC:**true`
- `**VPC:**false`

If no DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be publicly accessible. If a specific DB subnet group has been specified as part of the request and the `PubliclyAccessible` value has not been set, the DB instance will be private.

Parameters

- **auto_minor_version_upgrade** (*boolean*) – Indicates that minor version upgrades will be applied automatically to the DB instance during the maintenance window.
- **license_model** (*string*) – License model information for the restored DB instance.

Default: Same as source.

Valid values: `license-included` | `bring-your-own-license` | `'general-public-license'`

Parameters `db_name` (*string*) –

The database name for the restored DB instance.

This parameter is not used for the MySQL engine.

Parameters `engine` (*string*) – The database engine to use for the new instance.

Default: The same as source

Constraint: Must be compatible with the engine of the source

Example: `oracle-ee`

Parameters `iops` (*integer*) – The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.

Constraints: Must be an integer greater than 1000.

Parameters `option_group_name` (*string*) – The name of the option group to be used for the restored DB instance.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, cannot be removed from an option group, and that option group cannot be removed from a DB instance once it is associated with a DB instance

Parameters `tags` (*list*) – A list of tags. Tags must be passed as tuples in the form `[('key1', 'valueForKey1'), ('key2', 'valueForKey2')]`

revoke_db_security_group_ingress (*db_security_group_name*, *cidrip=None*,
ec2_security_group_name=None,
ec2_security_group_id=None,
ec2_security_group_owner_id=None)

Revokes ingress from a DBSecurityGroup for previously authorized IP ranges or EC2 or VPC Security Groups. Required parameters for this API are one of CIDRIP, EC2SecurityGroupId for VPC, or (EC2SecurityGroupOwnerId and either EC2SecurityGroupName or EC2SecurityGroupId).

Parameters

- **db_security_group_name** (*string*) – The name of the DB security group to revoke ingress from.
- **cidrip** (*string*) – The IP range to revoke access from. Must be a valid CIDR range. If *CIDRIP* is specified, *EC2SecurityGroupName*, *EC2SecurityGroupId* and *EC2SecurityGroupOwnerId* cannot be provided.
- **ec2_security_group_name** (*string*) – The name of the EC2 security group to revoke access from. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.
- **ec2_security_group_id** (*string*) – The id of the EC2 security group to revoke access from. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.
- **ec2_security_group_owner_id** (*string*) – The AWS Account Number of the owner of the EC2 security group specified in the *EC2SecurityGroupName* parameter. The AWS Access Key ID is not an acceptable value. For VPC DB security groups, *EC2SecurityGroupId* must be provided. Otherwise, *EC2SecurityGroupOwnerId* and either *EC2SecurityGroupName* or *EC2SecurityGroupId* must be provided.

4.144 Route 53 Domains

4.144.1 boto.route53.domains

```
boto.route53.domains.connect_to_region(region_name, **kw_params)
```

```
boto.route53.domains.regions()
```

Get all available regions for the Amazon Route 53 Domains service. :rtype: list :return: A list of boto.regioninfo.RegionInfo

4.144.2 boto.route53.domains.layer1

```
class boto.route53.domains.layer1.Route53DomainsConnection(**kwargs)
```

```
    APIVersion = '2014-05-15'
```

```
    DefaultRegionEndpoint = 'route53domains.us-east-1.amazonaws.com'
```

```
    DefaultRegionName = 'us-east-1'
```

```
    ResponseError
```

```
        alias of boto.exception.JSONResponseError
```

```
    ServiceName = 'Route53Domains'
```

```
    TargetPrefix = 'Route53Domains_v20140515'
```

```
    check_domain_availability(domain_name, idn_lang_code=None)
```

This operation checks the availability of one domain name. You can access this API without authenticating. Note that if the availability status of a domain is pending, you must submit another request to determine the availability of the domain name.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters `idn_lang_code` (*string*) – Reserved for future use.

disable_domain_transfer_lock (*domain_name*)

This operation removes the transfer lock on the domain (specifically the *clientTransferProhibited* status) to allow domain transfers. We recommend you refrain from performing this action unless you intend to transfer the domain to a different registrar. Successful submission returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

enable_domain_transfer_lock (*domain_name*)

This operation sets the transfer lock on the domain (specifically the *clientTransferProhibited* status) to prevent domain transfers. Successful submission returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

get_domain_detail (*domain_name*)

This operation returns detailed information about the domain. The domain's contact information is also returned as part of the output.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

get_operation_detail (*operation_id*)

This operation returns the current status of an operation that is not completed.

Parameters `operation_id` (*string*) – The identifier for the operation for which you want to get the status. Amazon Route 53 returned the identifier in the response to the original request.

Type: String

Default: None

Required: Yes

list_domains (*marker=None, max_items=None*)

This operation returns all the domain names registered with Amazon Route 53 for the current AWS account.

Parameters **marker** (*string*) – For an initial request for a list of domains, omit this element. If the number of domains that are associated with the current AWS account is greater than the value that you specified for *MaxItems*, you can use *Marker* to return additional domains. Get the value of *NextPageMarker* from the previous response, and submit another request that includes the value of *NextPageMarker* in the *Marker* element.

Type: String

Default: None

Constraints: The marker must match the value specified in the previous request.

Required: No

Parameters **max_items** (*integer*) – Number of domains to be returned.

Type: Integer

Default: 20

Constraints: A numeral between 1 and 100.

Required: No

list_operations (*marker=None, max_items=None*)

This operation returns the operation IDs of operations that are not yet complete.

Parameters **marker** (*string*) – For an initial request for a list of operations, omit this element. If the number of operations that are not yet complete is greater than the value that you specified for *MaxItems*, you can use *Marker* to return additional operations. Get the value of *NextPageMarker* from the previous response, and submit another request that includes the value of *NextPageMarker* in the *Marker* element.

Type: String

Default: None

Required: No

Parameters **max_items** (*integer*) – Number of domains to be returned.

Type: Integer

Default: 20

Constraints: A value between 1 and 100.

Required: No

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

register_domain (*domain_name, duration_in_years, admin_contact, registrant_contact, tech_contact, idn_lang_code=None, auto_renew=None, privacy_protect_admin_contact=None, privacy_protect_registrant_contact=None, privacy_protect_tech_contact=None*)

This operation registers a domain. Domains are registered by the AWS registrar partner, Gandi. For some top-level domains (TLDs), this operation requires extra parameters.

When you register a domain, Amazon Route 53 does the following:

- Creates a Amazon Route 53 hosted zone that has the same name as the domain. Amazon Route 53 assigns four name servers to your hosted zone and automatically updates your domain registration with the names of these name servers.
- Enables autorenew, so your domain registration will renew automatically each year. We'll notify you in advance of the renewal date so you can choose whether to renew the registration.
- Optionally enables privacy protection, so WHOIS queries return contact information for our registrar partner, Gandi, instead of the information you entered for registrant, admin, and tech contacts.

- If registration is successful, returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant is notified by email.
- Charges your AWS account an amount based on the top-level domain. For more information, see [‘Amazon Route 53 Pricing’](#).

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters

- `idn_lang_code` (*string*) – Reserved for future use.
- `duration_in_years` (*integer*) – The number of years the domain will be registered. Domains are registered for a minimum of one year. The maximum period depends on the top-level domain.

Type: Integer

Default: 1

Valid values: Integer from 1 to 10

Required: Yes

Parameters `auto_renew` (*boolean*) – Indicates whether the domain will be automatically renewed (*True*) or not (*False*). Autorenewal only takes effect after the account is charged.

Type: Boolean

Valid values: *True* | *False*

Default: *True*

Required: No

Parameters `admin_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters `registrant_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters `tech_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters `privacy_protect_admin_contact` (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

Parameters `privacy_protect_registrant_contact` (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

Parameters `privacy_protect_tech_contact` (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

retrieve_domain_auth_code (*domain_name*)

This operation returns the AuthCode for the domain. To transfer a domain to another registrar, you provide this value to the new registrar.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

transfer_domain (*domain_name*, *duration_in_years*, *nameservers*, *admin_contact*, *registrant_contact*, *tech_contact*, *idn_lang_code=None*, *auth_code=None*, *auto_renew=None*, *privacy_protect_admin_contact=None*, *privacy_protect_registrant_contact=None*, *privacy_protect_tech_contact=None*)

This operation transfers a domain from another registrar to Amazon Route 53. Domains are registered by the AWS registrar, Gandi upon transfer.

To transfer a domain, you need to meet all the domain transfer criteria, including the following:

- You must supply nameservers to transfer a domain.
- You must disable the domain transfer lock (if any) before transferring the domain.
- A minimum of 60 days must have elapsed since the domain’s registration or last transfer.

We recommend you use the Amazon Route 53 as the DNS service for your domain. You can create a hosted zone in Amazon Route 53 for your current domain before transferring your domain.

Note that upon transfer, the domain duration is extended for a year if not otherwise specified. Autorenew is enabled by default.

If the transfer is successful, this method returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Transferring domains charges your AWS account an amount based on the top-level domain. For more information, see [‘Amazon Route 53 Pricing’](#).

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters

- **idn_lang_code** (*string*) – Reserved for future use.
- **duration_in_years** (*integer*) – The number of years the domain will be registered. Domains are registered for a minimum of one year. The maximum period depends on the top-level domain.

Type: Integer

Default: 1

Valid values: Integer from 1 to 10

Required: Yes

Parameters nameservers (*list*) – Contains details for the host and glue IP addresses.

Type: Complex

Children: *GlueIps, Name*

Parameters auth_code (*string*) – The authorization code for the domain. You get this value from the current registrar.

Type: String

Required: Yes

Parameters auto_renew (*boolean*) – Indicates whether the domain will be automatically renewed (true) or not (false). Autorenewal only takes effect after the account is charged.

Type: Boolean

Valid values: *True | False*

Default: true

Required: No

Parameters admin_contact (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters registrant_contact (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters tech_contact (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName, MiddleName, LastName, ContactType, OrganizationName, AddressLine1, AddressLine2, City, State, CountryCode, ZipCode, PhoneNumber, Email, Fax, ExtraParams*

Required: Yes

Parameters privacy_protect_admin_contact (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

Parameters `privacy_protect_registrant_contact` (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

Parameters `privacy_protect_tech_contact` (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: *True*

Valid values: *True* | *False*

Required: No

`update_domain_contact` (*domain_name*, *admin_contact=None*, *registrant_contact=None*, *tech_contact=None*)

This operation updates the contact information for a particular domain. Information for at least one contact (registrant, administrator, or technical) must be supplied for update.

If the update is successful, this method returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Parameters `domain_name` (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters `admin_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName*, *MiddleName*, *LastName*, *ContactType*, *OrganizationName*, *AddressLine1*, *AddressLine2*, *City*, *State*, *CountryCode*, *ZipCode*, *PhoneNumber*, *Email*, *Fax*, *ExtraParams*

Required: Yes

Parameters `registrant_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName*, *MiddleName*, *LastName*, *ContactType*, *OrganizationName*, *AddressLine1*, *AddressLine2*, *City*, *State*, *CountryCode*, *ZipCode*, *PhoneNumber*, *Email*, *Fax*, *ExtraParams*

Required: Yes

Parameters `tech_contact` (*dict*) – Provides detailed contact information.

Type: Complex

Children: *FirstName*, *MiddleName*, *LastName*, *ContactType*, *OrganizationName*, *AddressLine1*, *AddressLine2*, *City*, *State*, *CountryCode*, *ZipCode*, *PhoneNumber*, *Email*, *Fax*, *ExtraParams*

Required: Yes

update_domain_contact_privacy (*domain_name*, *admin_privacy=None*, *registrant_privacy=None*, *tech_privacy=None*)

This operation updates the specified domain contact's privacy setting. When the privacy option is enabled, personal information such as postal or email address is hidden from the results of a public WHOIS query. The privacy services are provided by the AWS registrar, Gandi. For more information, see the '[Gandi privacy features](#)'.

This operation only affects the privacy of the specified contact type (registrant, administrator, or tech). Successful acceptance returns an operation ID that you can use with `GetOperationDetail` to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Parameters **domain_name** (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters **admin_privacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: *True* | *False*

Required: No

Parameters **registrant_privacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: *True* | *False*

Required: No

Parameters **tech_privacy** (*boolean*) – Whether you want to conceal contact information from WHOIS queries. If you specify true, WHOIS (“who is”) queries will return contact information for our registrar partner, Gandi, instead of the contact information that you enter.

Type: Boolean

Default: None

Valid values: *True* | *False*

Required: No

update_domain_nameservers (*domain_name*, *nameservers*)

This operation replaces the current set of name servers for the domain with the specified set of name servers. If you use Amazon Route 53 as your DNS service, specify the four name servers in the delegation set for the hosted zone for the domain.

If successful, this operation returns an operation ID that you can use to track the progress and completion of the action. If the request is not completed successfully, the domain registrant will be notified by email.

Parameters **domain_name** (*string*) – The name of a domain.

Type: String

Default: None

Constraints: The domain name can contain only the letters a through z, the numbers 0 through 9, and hyphen (-). Internationalized Domain Names are not supported.

Required: Yes

Parameters **nameservers** (*list*) – A list of new name servers for the domain.

Type: Complex

Children: *Name*, *GlueIps*

Required: Yes

4.144.3 boto.route53.domains.exceptions

```
exception boto.route53.domains.exceptions.DomainLimitExceeded (status, reason,  
                                                             body=None,  
                                                             *args)
```

```
exception boto.route53.domains.exceptions.DuplicateRequest (status, reason,  
                                                             body=None, *args)
```

```
exception boto.route53.domains.exceptions.InvalidInput (status, reason, body=None,  
                                                         *args)
```

```
exception boto.route53.domains.exceptions.OperationLimitExceeded (status,  
                                                                    reason,  
                                                                    body=None,  
                                                                    *args)
```

```
exception boto.route53.domains.exceptions.TLDRulesViolation (status, reason,  
                                                             body=None, *args)
```

```
exception boto.route53.domains.exceptions.UnsupportedTLD (status, reason,  
                                                            body=None, *args)
```

4.145 SDB DB Reference

This module offers an ORM-like layer on top of SimpleDB.

4.145.1 boto.sdb.db

4.145.2 boto.sdb.db.blob

```
class boto.sdb.db.blob.Blob (value=None, file=None, id=None)
```

Blob object

file

next ()

read ()

readline ()

size

4.145.3 boto.sdb.db.key

class boto.sdb.db.key.**Key** (*encoded=None, obj=None*)

```

app ()
classmethod from_path (*args, **kws)
has_id_or_name ()
id ()
id_or_name ()
kind ()
name ()
parent ()

```

4.145.4 boto.sdb.db.manager

boto.sdb.db.manager.**get_manager** (*cls*)

Returns the appropriate Manager class for a given Model class. It does this by looking in the boto config for a section like this:

```

[DB]
db_type = SimpleDB
db_user = <aws access key id>
db_passwd = <aws secret access key>
db_name = my_domain
[DB_TestBasic]
db_type = SimpleDB
db_user = <another aws access key id>
db_passwd = <another aws secret access key>
db_name = basic_domain
db_port = 1111

```

The values in the DB section are “generic values” that will be used if nothing more specific is found. You can also create a section for a specific Model class that gives the db info for that class. In the example above, TestBasic is a Model subclass.

4.145.5 boto.sdb.db.manager.sdbmanager

class boto.sdb.db.manager.sdbmanager.**SDBConverter** (*manager*)

Responsible for converting base Python types to format compatible with underlying database. For SimpleDB, that means everything needs to be converted to a string when stored in SimpleDB and from a string when retrieved.

To convert a value, pass it to the encode or decode method. The encode method will take a Python native value and convert to DB format. The decode method will take a DB format value and convert it to Python native format. To find the appropriate method to call, the generic encode/decode methods will look for the type-specific method by searching for a method called “encode_<type name>” or “decode_<type name>”.

```

decode (item_type, value)
decode_blob (value)

```

decode_bool (*value*)

decode_date (*value*)

decode_datetime (*value*)

Handles both Dates and DateTime objects

decode_float (*value*)

decode_int (*value*)

decode_list (*prop, value*)

decode_long (*value*)

decode_map (*prop, value*)

decode_map_element (*item_type, value*)

Decode a single element for a map

decode_prop (*prop, value*)

decode_reference (*value*)

decode_string (*value*)

Decoding a string is really nothing, just return the value as-is

decode_time (*value*)

converts strings in the form of HH:MM:SS.mmmmmm (created by `datetime.time.isoformat()`) to `datetime.time` objects.

Timzone-aware strings (“HH:MM:SS.mmmmmm+HH:MM”) won’t be handled right now and will raise `TimeDecodeError`.

encode (*item_type, value*)

encode_blob (*value*)

encode_bool (*value*)

encode_date (*value*)

encode_datetime (*value*)

encode_float (*value*)

See <http://tools.ietf.org/html/draft-wood-ldapext-float-00>.

encode_int (*value*)

encode_list (*prop, value*)

encode_long (*value*)

encode_map (*prop, value*)

encode_prop (*prop, value*)

encode_reference (*value*)

encode_string (*value*)

Convert ASCII, Latin-1 or UTF-8 to pure Unicode

encode_time (*value*)

class `boto.sdb.db.manager.sdbmanager.SDBManager` (*cls, db_name, db_user, db_passwd, db_host, db_port, db_table, ddl_dir, enable_ssl, consistent=None*)

```

count (cls, filters, quick=True, sort_by=None, select=None)
    Get the number of results that would be returned in this query

decode_value (prop, value)

delete_key_value (obj, name)

delete_object (obj)

domain

encode_value (prop, value)

get_blob_bucket (bucket_name=None)

get_key_value (obj, name)

get_object (cls, id, a=None)

get_object_from_id (id)

get_property (prop, obj, name)

get_raw_item (obj)

get_s3_connection ()

load_object (obj)

query (query)

query_gql (query_string, *args, **kws)

save_object (obj, expected_value=None)

sdb

set_key_value (obj, name, value)

set_property (prop, obj, name, value)

```

```
exception boto.sdb.db.manager.sdbmanager.TimeDecodeError
```

4.145.6 boto.sdb.db.manager.xmlmanager

4.145.7 boto.sdb.db.model

4.145.8 boto.sdb.db.property

```
class boto.sdb.db.property.BlobProperty (verbose_name=None, name=None, default=None,
                                         required=False, validator=None, choices=None,
                                         unique=False)
```

```

data_type
    alias of boto.sdb.db.blob.Blob

```

```
type_name = 'blob'
```

```
class boto.sdb.db.property.BooleanProperty (verbose_name=None, name=None, de-
                                             fault=False, required=False, validator=None,
                                             choices=None, unique=False)
```

```

data_type
    alias of __builtin__.bool

```

```
empty (value)  
type_name = 'Boolean'  
class boto.sdb.db.property.CalculatedProperty (verbose_name=None, name=None, default=None, required=False, validator=None, choices=None, calculated_type=<type 'int'>, unique=False, use_method=False)
```

```
get_value_for_datastore (model_instance)
```

```
class boto.sdb.db.property.DateProperty (verbose_name=None, auto_now=False, auto_now_add=False, name=None, default=None, required=False, validator=None, choices=None, unique=False)
```

```
data_type  
    alias of datetime.date
```

```
default_value ()
```

```
get_value_for_datastore (model_instance)
```

```
now ()
```

```
type_name = 'Date'
```

```
validate (value)
```

```
class boto.sdb.db.property.DateTimeProperty (verbose_name=None, auto_now=False, auto_now_add=False, name=None, default=None, required=False, validator=None, choices=None, unique=False)
```

This class handles both the `datetime.datetime` object And the `datetime.date` objects. It can return either one, depending on the value stored in the database

```
data_type  
    alias of datetime.datetime
```

```
default_value ()
```

```
get_value_for_datastore (model_instance)
```

```
now ()
```

```
type_name = 'DateTime'
```

```
validate (value)
```

```
class boto.sdb.db.property.FloatProperty (verbose_name=None, name=None, default=0.0, required=False, validator=None, choices=None, unique=False)
```

```
data_type  
    alias of __builtin__.float
```

```
empty (value)
```

```
type_name = 'Float'
```

```
validate (value)
```



```
class boto.sdb.db.property.IntegerProperty (verbose_name=None, name=None,
default=0, required=False, validator=None, choices=None, unique=False,
max=2147483647, min=-2147483648)
```

```
data_type
    alias of __builtin__.int
```

```
empty (value)
```

```
type_name = 'Integer'
```

```
validate (value)
```

```
class boto.sdb.db.property.ListProperty (item_type, verbose_name=None, name=None, de-
fault=None, **kwds)
```

```
data_type
    alias of __builtin__.list
```

```
default_value ()
```

```
empty (value)
```

```
type_name = 'List'
```

```
validate (value)
```

```
class boto.sdb.db.property.LongProperty (verbose_name=None, name=None, default=0, re-
quired=False, validator=None, choices=None,
unique=False)
```

```
data_type
    alias of __builtin__.long
```

```
empty (value)
```

```
type_name = 'Long'
```

```
validate (value)
```

```
class boto.sdb.db.property.MapProperty (item_type=<type 'str'>, verbose_name=None,
name=None, default=None, **kwds)
```

```
data_type
    alias of __builtin__.dict
```

```
default_value ()
```

```
empty (value)
```

```
type_name = 'Map'
```

```
validate (value)
```

```
class boto.sdb.db.property.PasswordProperty (verbose_name=None, name=None, de-
fault="", required=False, validator=None,
choices=None, unique=False, hash-
func=None)
```

Hashed property whose original value can not be retrieved, but still can be compared.

Works by storing a hash of the original value instead of the original value. Once that's done all that can be retrieved is the hash.

The comparison

```
obj.password == 'foo'
```

generates a hash of 'foo' and compares it to the stored hash.

Underlying data type for hashing, storing, and comparing is `boto.utils.Password`. The default hash function is defined there (currently sha512 in most cases, md5 where sha512 is not available)

It's unlikely you'll ever need to use a different hash function, but if you do, you can control the behavior in one of two ways:

1. Specifying hashfunc in PasswordProperty constructor

```
import hashlib
class MyModel(model): password = PasswordProperty(hashfunc=hashlib.sha224)
```

2. Subclassing Password and PasswordProperty

```
class SHA224Password(Password): hashfunc=hashlib.sha224
class SHA224PasswordProperty(PasswordProperty): data_type=MyPassword
    type_name="MyPassword"
class MyModel(Model): password = SHA224PasswordProperty()
```

The hashfunc parameter overrides the default hashfunc in `boto.utils.Password`.

The remaining parameters are passed through to `StringProperty.__init__`

data_type

alias of `boto.utils.Password`

get_value_for_datastore (*model_instance*)

make_value_from_datastore (*value*)

type_name = 'Password'

validate (*value*)

```
class boto.sdb.db.property.Property(verbose_name=None, name=None, default=None,
                                   required=False, validator=None, choices=None,
                                   unique=False)
```

data_type

alias of `__builtin__.str`

default_validator (*value*)

default_value ()

empty (*value*)

get_choices ()

get_value_for_datastore (*model_instance*)

make_value_from_datastore (*value*)

name = ''

type_name = ''

validate (*value*)

verbose_name = ''

```
class boto.sdb.db.property.ReferenceProperty(reference_class=None, collection_name=None,
                                              verbose_name=None, name=None, default=None,
                                              required=False, validator=None, choices=None,
                                              unique=False)
```

```

check_instance (value)
check_uuid (value)
data_type
    alias of boto.sdb.db.key.Key
type_name = 'Reference'
validate (value)
class boto.sdb.db.property.S3KeyProperty (verbose_name=None, name=None, default=None, required=False, validator=None, choices=None, unique=False)

data_type
    alias of boto.s3.key.Key
get_value_for_datastore (model_instance)
type_name = 'S3Key'
validate (value)
validate_regex = '^s3:\\/\\/\\/([^\\/]*)\\/(.*)$'
class boto.sdb.db.property.StringProperty (verbose_name=None, name=None, default="", required=False, validator=<function validate_string>, choices=None, unique=False)

type_name = 'String'
class boto.sdb.db.property.TextProperty (verbose_name=None, name=None, default="", required=False, validator=None, choices=None, unique=False, max_length=None)

type_name = 'Text'
validate (value)
class boto.sdb.db.property.TimeProperty (verbose_name=None, name=None, default=None, required=False, validator=None, choices=None, unique=False)

data_type
    alias of datetime.time
type_name = 'Time'
validate (value)
boto.sdb.db.property.validate_string (value)

```

4.145.9 boto.sdb.db.query

```

class boto.sdb.db.query.Query (model_class, limit=None, next_token=None, manager=None)

count (quick=True)
fetch (limit, offset=0)
    Not currently fully supported, but we can use this to allow them to set a limit in a chainable method
filter (property_operator, value)

```

```
get_next_token ()
get_query ()
next ()
next_token
order (key)
set_next_token (token)
to_xml (doc=None)
```

4.146 Support

4.146.1 boto.support

```
boto.support.connect_to_region (region_name, **kw_params)
```

```
boto.support.regions ()
```

Get all available regions for the Amazon Support service.

Return type list

Returns A list of `boto.regioninfo.RegionInfo`

4.146.2 boto.support.layer1

```
class boto.support.layer1.SupportConnection (**kwargs)
```

AWS Support The AWS Support API reference is intended for programmers who need detailed information about the AWS Support operations and data types. This service enables you to manage your AWS Support cases programmatically. It uses HTTP methods that return results in JSON format.

The AWS Support service also exposes a set of ‘**Trusted Advisor**’_ features. You can retrieve a list of checks and their descriptions, get check results, specify checks to refresh, and get the refresh status of checks.

The following list describes the AWS Support case management operations:

- ****Service names, issue categories, and available severity levels.** ******The DescribeServices and DescribeSeverityLevels operations return AWS service names, service codes, service categories, and problem severity levels. You use these values when you call the CreateCase operation.
- **Case creation, case details, and case resolution.** The CreateCase, DescribeCases, DescribeAttachment, and ResolveCase operations create AWS Support cases, retrieve information about cases, and resolve cases.
- **Case communication.** The DescribeCommunications, AddCommunicationToCase, and AddAttachmentsToSet operations retrieve and add communications and attachments to AWS Support cases.

The following list describes the operations available from the AWS Support service for Trusted Advisor:

- DescribeTrustedAdvisorChecks returns the list of checks that run against your AWS resources.
- Using the *CheckId* for a specific check returned by DescribeTrustedAdvisorChecks, you can call DescribeTrustedAdvisorCheckResult to obtain the results for the check you specified.
- DescribeTrustedAdvisorCheckSummaries returns summarized results for one or more Trusted Advisor checks.
- RefreshTrustedAdvisorCheck requests that Trusted Advisor rerun a specified check.
- DescribeTrustedAdvisorCheckRefreshStatuses reports the refresh status of one or more checks.

For authentication of requests, AWS Support uses ‘**Signature Version 4 Signing Process**’_.

See [‘About the AWS Support API’](#) in the AWS Support User Guide for information about how to use this service to create and manage your support cases, and how to call Trusted Advisor for results of checks on your resources.

```
APIVersion = '2013-04-15'
```

```
DefaultRegionEndpoint = 'support.us-east-1.amazonaws.com'
```

```
DefaultRegionName = 'us-east-1'
```

```
ResponseError
```

alias of `boto.exception.JSONResponseError`

```
ServiceName = 'Support'
```

```
TargetPrefix = 'AWSSupport_20130415'
```

```
add_attachments_to_set (attachments, attachment_set_id=None)
```

Adds one or more attachments to an attachment set. If an *AttachmentSetId* is not specified, a new attachment set is created, and the ID of the set is returned in the response. If an *AttachmentSetId* is specified, the attachments are added to the specified set, if it exists.

An attachment set is a temporary container for attachments that are to be added to a case or case communication. The set is available for one hour after it is created; the *ExpiryTime* returned in the response indicates when the set expires. The maximum number of attachments in a set is 3, and the maximum size of any attachment in the set is 5 MB.

Parameters

- **attachment_set_id** (*string*) – The ID of the attachment set. If an *AttachmentSetId* is not specified, a new attachment set is created, and the ID of the set is returned in the response. If an *AttachmentSetId* is specified, the attachments are added to the specified set, if it exists.
- **attachments** (*list*) – One or more attachments to add to the set. The limit is 3 attachments per set, and the size limit is 5 MB per attachment.

```
add_communication_to_case (communication_body, case_id=None,
                           cc_email_addresses=None, attachment_set_id=None)
```

Adds additional customer communication to an AWS Support case. You use the *CaseId* value to identify the case to add communication to. You can list a set of email addresses to copy on the communication using the *CcEmailAddresses* value. The *CommunicationBody* value contains the text of the communication.

The response indicates the success or failure of the request.

This operation implements a subset of the behavior on the AWS Support [‘Your Support Cases’](#) web form.

Parameters

- **case_id** (*string*) – The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case-12345678910-2013-c4c1d2bf33c5cf47
- **communication_body** (*string*) – The body of an email communication to add to the support case.
- **cc_email_addresses** (*list*) – The email addresses in the CC line of an email to be added to the support case.
- **attachment_set_id** (*string*) – The ID of a set of one or more attachments for the communication to add to the case. Create the set by calling `AddAttachmentsToSet`

create_case (*subject*, *communication_body*, *service_code=None*, *severity_code=None*, *category_code=None*, *cc_email_addresses=None*, *language=None*, *issue_type=None*, *attachment_set_id=None*)

Creates a new case in the AWS Support Center. This operation is modeled on the behavior of the AWS Support Center **‘Open a new case’** page. Its parameters require you to specify the following information:

1. **IssueType.** The type of issue for the case. You can specify either “customer-service” or “technical.” If you do not indicate a value, the default is “technical.”
2. **ServiceCode.** The code for an AWS service. You obtain the *ServiceCode* by calling *DescribeServices*.
3. **CategoryCode.** The category for the service defined for the *ServiceCode* value. You also obtain the category code for a service by calling *DescribeServices*. Each AWS service defines its own set of category codes.
4. **SeverityCode.** A value that indicates the urgency of the case, which in turn determines the response time according to your service level agreement with AWS Support. You obtain the *SeverityCode* by calling *DescribeSeverityLevels*.
5. **Subject.** The **Subject** field on the AWS Support Center **‘Open a new case’** page.
6. **CommunicationBody.** The **Description** field on the AWS Support Center **‘Open a new case’** page.
7. **AttachmentSetId.** The ID of a set of attachments that has been created by using *AddAttachmentsToSet*.
8. **Language.** The human language in which AWS Support handles the case. English and Japanese are currently supported.
9. **CcEmailAddresses.** The AWS Support Center **CC** field on the **‘Open a new case’** page. You can list email addresses to be copied on any correspondence about the case. The account that opens the case is already identified by passing the AWS Credentials in the HTTP POST method or in a method or function call from one of the programming languages supported by an **‘AWS SDK’**.

A successful *CreateCase* request returns an AWS Support case number. Case numbers are used by the *DescribeCases* operation to retrieve existing AWS Support cases.

Parameters

- **subject** (*string*) – The title of the AWS Support case.
- **service_code** (*string*) – The code for the AWS service returned by the call to *DescribeServices*.
- **severity_code** (*string*) – The code for the severity level returned by the call to *DescribeSeverityLevels*.
- **category_code** (*string*) – The category of problem for the AWS Support case.
- **communication_body** (*string*) – The communication body text when you create an AWS Support case by calling *CreateCase*.
- **cc_email_addresses** (*list*) – A list of email addresses that AWS Support copies on case correspondence.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **issue_type** (*string*) – The type of issue for the case. You can specify either “customer-service” or “technical.” If you do not indicate a value, the default is “technical.”
- **attachment_set_id** (*string*) – The ID of a set of one or more attachments for the case. Create the set by using *AddAttachmentsToSet*.

describe_attachment (*attachment_id*)

Returns the attachment that has the specified ID. Attachment IDs are generated by the case management system when you add an attachment to a case or case communication. Attachment IDs are returned in the AttachmentDetails objects that are returned by the DescribeCommunications operation.

Parameters **attachment_id** (*string*) – The ID of the attachment to return. Attachment IDs are returned by the DescribeCommunications operation.

describe_cases (*case_id_list=None, display_id=None, after_time=None, before_time=None, include_resolved_cases=None, next_token=None, max_results=None, language=None, include_communications=None*)

Returns a list of cases that you specify by passing one or more case IDs. In addition, you can filter the cases by date by setting values for the *AfterTime* and *BeforeTime* request parameters.

Case data is available for 12 months after creation. If a case was created more than 12 months ago, a request for data might cause an error.

The response returns the following in JSON format:

1. One or more CaseDetails data types.
2. One or more NextToken values, which specify where to paginate the returned records represented by the CaseDetails objects.

Parameters

- **case_id_list** (*list*) – A list of ID numbers of the support cases you want returned. The maximum number of cases is 100.
- **display_id** (*string*) – The ID displayed for a case in the AWS Support Center user interface.
- **after_time** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **before_time** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **include_resolved_cases** (*boolean*) – Specifies whether resolved support cases should be included in the DescribeCases results. The default is false.
- **next_token** (*string*) – A resumption point for pagination.
- **max_results** (*integer*) – The maximum number of results to return before paginating.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.
- **include_communications** (*boolean*) – Specifies whether communications should be included in the DescribeCases results. The default is true.

describe_communications (*case_id, before_time=None, after_time=None, next_token=None, max_results=None*)

Returns communications (and attachments) for one or more support cases. You can use the *AfterTime* and *BeforeTime* parameters to filter by date. You can use the *CaseId* parameter to restrict the results to a particular case.

Case data is available for 12 months after creation. If a case was created more than 12 months ago, a request for data might cause an error.

You can use the *MaxResults* and *NextToken* parameters to control the pagination of the result set. Set *MaxResults* to the number of cases you want displayed on each page, and use *NextToken* to specify the resumption of pagination.

Parameters

- **case_id** (*string*) – The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case- 12345678910-2013-c4c1d2bf33c5cf47
- **before_time** (*string*) – The end date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **after_time** (*string*) – The start date for a filtered date search on support case communications. Case communications are available for 12 months after creation.
- **next_token** (*string*) – A resumption point for pagination.
- **max_results** (*integer*) – The maximum number of results to return before paginating.

describe_services (*service_code_list=None, language=None*)

Returns the current list of AWS services and a list of service categories that applies to each one. You then use service names and categories in your CreateCase requests. Each AWS service has its own set of categories.

The service codes and category codes correspond to the values that are displayed in the **Service** and **Category** drop-down lists on the AWS Support Center ‘**Open a new case**’ page. The values in those fields, however, do not necessarily match the service codes and categories returned by the *DescribeServices* request. Always use the service codes and categories obtained programmatically. This practice ensures that you always have the most recent set of service and category codes.

Parameters

- **service_code_list** (*list*) – A JSON-formatted list of service codes available for AWS services.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

describe_severity_levels (*language=None*)

Returns the list of severity levels that you can assign to an AWS Support case. The severity level for a case is also a field in the CaseDetails data type included in any CreateCase request.

Parameters **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

describe_trusted_advisor_check_refresh_statuses (*check_ids*)

Returns the refresh status of the Trusted Advisor checks that have the specified check IDs. Check IDs can be obtained by calling DescribeTrustedAdvisorChecks.

Parameters **check_ids** (*list*) – The IDs of the Trusted Advisor checks.

describe_trusted_advisor_check_result (*check_id, language=None*)

Returns the results of the Trusted Advisor check that has the specified check ID. Check IDs can be obtained by calling DescribeTrustedAdvisorChecks.

The response contains a TrustedAdvisorCheckResult object, which contains these three objects:

- TrustedAdvisorCategorySpecificSummary

- `TrustedAdvisorResourceDetail`
- `TrustedAdvisorResourcesSummary`

In addition, the response contains these fields:

- **Status.** The alert status of the check: “ok” (green), “warning” (yellow), “error” (red), or “not_available”.
- **Timestamp.** The time of the last refresh of the check.
- **CheckId.** The unique identifier for the check.

Parameters

- **check_id** (*string*) – The unique identifier for the Trusted Advisor check.
- **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

describe_trusted_advisor_check_summaries (*check_ids*)

Returns the summaries of the results of the Trusted Advisor checks that have the specified check IDs. Check IDs can be obtained by calling `DescribeTrustedAdvisorChecks`.

The response contains an array of `TrustedAdvisorCheckSummary` objects.

Parameters **check_ids** (*list*) – The IDs of the Trusted Advisor checks.

describe_trusted_advisor_checks (*language*)

Returns information about all available Trusted Advisor checks, including name, ID, category, description, and metadata. You must specify a language code; English (“en”) and Japanese (“ja”) are currently supported. The response contains a `TrustedAdvisorCheckDescription` for each check.

Parameters **language** (*string*) – The ISO 639-1 code for the language in which AWS provides support. AWS Support currently supports English (“en”) and Japanese (“ja”). Language parameters must be passed explicitly for operations that take them.

make_request (*action, body*)

Makes a request to the server, with stock multiple-retry logic.

refresh_trusted_advisor_check (*check_id*)

Requests a refresh of the Trusted Advisor check that has the specified check ID. Check IDs can be obtained by calling `DescribeTrustedAdvisorChecks`.

The response contains a `RefreshTrustedAdvisorCheckResult` object, which contains these fields:

- **Status.** The refresh status of the check: “none”, “enqueued”, “processing”, “success”, or “abandoned”.
- **MillisUntilNextRefreshable.** The amount of time, in milliseconds, until the check is eligible for refresh.
- **CheckId.** The unique identifier for the check.

Parameters **check_id** (*string*) – The unique identifier for the Trusted Advisor check.

resolve_case (*case_id=None*)

Takes a *CaseId* and returns the initial state of the case along with the state of the case after the call to `ResolveCase` completed.

Parameters **case_id** (*string*) – The AWS Support case ID requested or returned in the call. The case ID is an alphanumeric string formatted as shown in this example: case-12345678910-2013-c4c1d2bf33c5cf47

4.146.3 boto.support.exceptions

exception `boto.support.exceptions.AttachmentIdNotFound` (*status, reason, body=None, *args*)

```
exception boto.support.exceptions.AttachmentLimitExceeded (status, reason,  
                                                    body=None, *args)  
exception boto.support.exceptions.AttachmentSetExpired (status, reason, body=None,  
                                                    *args)  
exception boto.support.exceptions.AttachmentSetIdNotFound (status, reason,  
                                                    body=None, *args)  
exception boto.support.exceptions.AttachmentSetSizeLimitExceeded (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.support.exceptions.CaseCreationLimitExceeded (status, reason,  
                                                    body=None, *args)  
exception boto.support.exceptions.CaseIdNotFound (status, reason, body=None, *args)  
exception boto.support.exceptions.DescribeAttachmentLimitExceeded (status,  
                                                    reason,  
                                                    body=None,  
                                                    *args)  
exception boto.support.exceptions.InternalServerError (status, reason, body=None,  
                                                    *args)
```

4.147 boto v2.46.0

date 2017/02/20

This release migrates boto2 to the new endpoint format which is also used by boto3. The major advantage this provides is the ability to connect to regions that aren't hard-coded in our built in endpoints file without having to go and find out what the hostname for that service/region is yourself. No more waiting for updates just to get region support!

Since this feature could potentially break assumptions, it is disabled by default. You can enable it by using either the `use_endpoint_heuristics` config variable or the `BOTO_USE_ENDPOINT_HEURISTICS` environment variable.

Even though we are changing the underlying format of our built in endpoints, the endpoints provided by `endpoints_path` or `BOTO_ENDPOINTS` will continue to use the legacy format, so you will not need to update your custom endpoints as part of this update.

4.147.1 Changes

- Endpoints v2 (issue 3675, commit d7253d8)

4.148 boto v2.xx.x

date 2013/xx/xx

This release adds ____.

4.148.1 Features

- . (:issue:, :sha:)

4.148.2 Bugfixes

- (:issue:, :sha:)
- Several documentation improvements/fixes:
 - (:issue:, :sha:)

4.149 boto v2.xx.x

date 2013/xx/xx

This release adds ____.

4.149.1 Features

- . (:issue:, :sha:)

4.149.2 Bugfixes

- (:issue:, :sha:)
- Several documentation improvements/fixes:
 - (:issue:, :sha:)

4.150 Indices and tables

- genindex
- modindex
- search

b

- boto, 204
- boto.awslambda, 188
- boto.awslambda.exceptions, 192
- boto.awslambda.layer1, 188
- boto.beanstalk, 192
- boto.beanstalk.layer1, 192
- boto.beanstalk.response, 202
- boto.cloudformation, 225
- boto.cloudformation.connection, 225
- boto.cloudformation.stack, 232
- boto.cloudformation.template, 234
- boto.cloudfront, 234
- boto.cloudfront.distribution, 235
- boto.cloudfront.exception, 243
- boto.cloudfront.identity, 240
- boto.cloudfront.invalidation, 241
- boto.cloudfront.logging, 243
- boto.cloudfront.object, 242
- boto.cloudfront.origin, 239
- boto.cloudfront.signers, 241
- boto.cloudhsm, 243
- boto.cloudhsm.exceptions, 246
- boto.cloudhsm.layer1, 243
- boto.cloudsearch, 246
- boto.cloudsearch.document, 260
- boto.cloudsearch.domain, 247
- boto.cloudsearch.layer1, 250
- boto.cloudsearch.layer2, 256
- boto.cloudsearch.optionstatus, 257
- boto.cloudsearch.search, 258
- boto.cloudsearch2, 262
- boto.cloudsearch2.document, 276
- boto.cloudsearch2.domain, 262
- boto.cloudsearch2.layer1, 266
- boto.cloudsearch2.layer2, 272
- boto.cloudsearch2.optionstatus, 272
- boto.cloudsearch2.search, 274
- boto.cloudsearchdomain, 278
- boto.cloudsearchdomain.exceptions, 283
- boto.cloudsearchdomain.layer1, 278
- boto.cloudtrail, 283
- boto.cloudtrail.exceptions, 286
- boto.cloudtrail.layer1, 284
- boto.codedeploy, 295
- boto.codedeploy.exceptions, 303
- boto.codedeploy.layer1, 296
- boto.cognito.identity, 306
- boto.cognito.identity.exceptions, 311
- boto.cognito.identity.layer1, 307
- boto.cognito.sync, 312
- boto.cognito.sync.exceptions, 316
- boto.cognito.sync.layer1, 312
- boto.configservice, 316
- boto.configservice.exceptions, 319
- boto.configservice.layer1, 316
- boto.connection, 213
- boto.contrib, 321
- boto.contrib.ymlmessage, 321
- boto.datapipeline, 321
- boto.datapipeline.exceptions, 327
- boto.datapipeline.layer1, 321
- boto.dynamodb, 327
- boto.dynamodb.batch, 345
- boto.dynamodb.item, 344
- boto.dynamodb.layer1, 327
- boto.dynamodb.layer2, 331
- boto.dynamodb.schema, 343
- boto.dynamodb.table, 338
- boto.dynamodb.types, 347
- boto.dynamodb2, 369
- boto.dynamodb2.exceptions, 402
- boto.dynamodb2.fields, 349
- boto.dynamodb2.items, 353
- boto.dynamodb2.layer1, 369
- boto.dynamodb2.results, 356
- boto.dynamodb2.table, 357
- boto.ec2, 403
- boto.ec2.address, 403

- boto.ec2.autoscale, 175
- boto.ec2.autoscale.activity, 182
- boto.ec2.autoscale.group, 182
- boto.ec2.autoscale.instance, 184
- boto.ec2.autoscale.launchconfig, 184
- boto.ec2.autoscale.policy, 186
- boto.ec2.autoscale.request, 187
- boto.ec2.autoscale.scheduled, 187
- boto.ec2.autoscale.tag, 187
- boto.ec2.blockdevicemapping, 404
- boto.ec2.buyreservation, 404
- boto.ec2.cloudwatch, 288
- boto.ec2.cloudwatch.alarm, 294
- boto.ec2.cloudwatch.datapoint, 292
- boto.ec2.cloudwatch.metric, 292
- boto.ec2.connection, 405
- boto.ec2.ec2object, 440
- boto.ec2.elb, 495
- boto.ec2.elb.attributes, 506
- boto.ec2.elb.healthcheck, 501
- boto.ec2.elb.instancestate, 501
- boto.ec2.elb.listelement, 502
- boto.ec2.elb.listener, 502
- boto.ec2.elb.loadbalancer, 502
- boto.ec2.elb.policies, 505
- boto.ec2.elb.securitygroup, 505
- boto.ec2.group, 441
- boto.ec2.image, 441
- boto.ec2.instance, 444
- boto.ec2.instanceinfo, 449
- boto.ec2.instancestatus, 450
- boto.ec2.keypair, 451
- boto.ec2.launchspecification, 451
- boto.ec2.networkinterface, 452
- boto.ec2.placementgroup, 454
- boto.ec2.regioninfo, 454
- boto.ec2.reservedinstance, 454
- boto.ec2.securitygroup, 456
- boto.ec2.snapshot, 457
- boto.ec2.spotinstancerequest, 458
- boto.ec2.tag, 460
- boto.ec2.volume, 460
- boto.ec2.volumestatus, 462
- boto.ec2.zone, 463
- boto.ec2containerservice, 464
- boto.ec2containerservice.exceptions, 469
- boto.ec2containerservice.layer1, 464
- boto.ecs, 470
- boto.ecs.item, 470
- boto.elasticache, 471
- boto.elasticache.layer1, 471
- boto.elastictranscoder, 485
- boto.elastictranscoder.exceptions, 495
- boto.elastictranscoder.layer1, 485
- boto.emr, 506
- boto.emr.connection, 506
- boto.emr.emrobject, 512
- boto.emr.step, 510
- boto.exception, 218
- boto.file.bucket, 515
- boto.file.connection, 516
- boto.file.key, 516
- boto.file.simpleresultset, 516
- boto.fps, 518
- boto.fps.connection, 518
- boto.glacier, 521
- boto.glacier.concurrent, 538
- boto.glacier.exceptions, 539
- boto.glacier.job, 536
- boto.glacier.layer1, 521
- boto.glacier.layer2, 533
- boto.glacier.vault, 533
- boto.glacier.writer, 537
- boto.gs.acl, 540
- boto.gs.bucket, 541
- boto.gs.bucketlistresultset, 555
- boto.gs.connection, 556
- boto.gs.cors, 559
- boto.gs.key, 559
- boto.gs.resumable_upload_handler, 572
- boto.gs.user, 572
- boto.handler, 220
- boto.iam, 573
- boto.iam.connection, 574
- boto.iam.summarymap, 588
- boto.kinesis, 588
- boto.kinesis.exceptions, 596
- boto.kinesis.layer1, 589
- boto.kms, 821
- boto.kms.exceptions, 828
- boto.kms.layer1, 822
- boto.logs, 828
- boto.logs.exceptions, 833
- boto.logs.layer1, 828
- boto.machinelearning, 833
- boto.machinelearning.exceptions, 846
- boto.machinelearning.layer1, 833
- boto.manage, 597
- boto.manage.cmdshell, 597
- boto.manage.propget, 600
- boto.mturk, 600
- boto.mturk.connection, 600
- boto.mturk.layoutparam, 605
- boto.mturk.notification, 605
- boto.mturk.price, 606
- boto.mturk.qualification, 606
- boto.mturk.question, 607
- boto.mws, 611

boto.mws.connection, 611
boto.mws.exception, 621
boto.mws.response, 621
boto.opsworks, 846
boto.opsworks.exceptions, 874
boto.opsworks.layer1, 847
boto.pyami, 630
boto.pyami.bootstrap, 630
boto.pyami.config, 630
boto.pyami.copybot, 631
boto.pyami.installers, 631
boto.pyami.installers.ubuntu, 632
boto.pyami.installers.ubuntu.apache, 632
boto.pyami.installers.ubuntu.installer, 632
boto.pyami.installers.ubuntu.mysql, 632
boto.pyami.installers.ubuntu.trac, 633
boto.pyami.launch_ami, 633
boto.pyami.scriptbase, 633
boto.pyami.startup, 633
boto.rds, 634
boto.rds.dbinstance, 648
boto.rds.dbsecuritygroup, 651
boto.rds.dbsnapshot, 652
boto.rds.event, 653
boto.rds.parametergroup, 653
boto.rds2, 874
boto.rds2.exceptions, 874
boto.rds2.layer1, 877
boto.redshift, 654
boto.redshift.exceptions, 681
boto.redshift.layer1, 654
boto.resultset, 220
boto.route53.connection, 688
boto.route53.domains, 910
boto.route53.domains.exceptions, 918
boto.route53.domains.layer1, 910
boto.route53.exception, 691
boto.route53.healthcheck, 691
boto.route53.hostedzone, 693
boto.route53.record, 693
boto.route53.status, 695
boto.route53.zone, 695
boto.s3, 697
boto.s3.acl, 697
boto.s3.bucket, 698
boto.s3.bucketlistresultset, 710
boto.s3.connection, 710
boto.s3.cors, 714
boto.s3.deletemarker, 716
boto.s3.key, 716
boto.s3.lifecycle, 730
boto.s3.multidelete, 728
boto.s3.multipart, 727
boto.s3.prefix, 727
boto.s3.resumable_download_handler, 729
boto.s3.tagging, 732
boto.s3.user, 732
boto.sdb, 733
boto.sdb.connection, 733
boto.sdb.db, 918
boto.sdb.db.blob, 918
boto.sdb.db.key, 919
boto.sdb.db.manager, 919
boto.sdb.db.manager.sdbmanager, 919
boto.sdb.db.property, 921
boto.sdb.db.query, 925
boto.sdb.domain, 738
boto.sdb.item, 741
boto.sdb.queryresultset, 742
boto.services, 743
boto.services.bs, 743
boto.services.message, 743
boto.services.result, 743
boto.services.service, 744
boto.services.servicedef, 744
boto.services.sonofmmm, 744
boto.services.submit, 745
boto.ses, 745
boto.ses.connection, 745
boto.sns, 749
boto.sqs, 754
boto.sqs.attributes, 755
boto.sqs.batchresults, 768
boto.sqs.connection, 755
boto.sqs.jsonmessage, 761
boto.sqs.message, 761
boto.sqs.queue, 763
boto.sqs.regioninfo, 768
boto.sts, 769
boto.sts.credentials, 775
boto.support, 926
boto.support.exceptions, 931
boto.support.layer1, 926
boto.swf, 777
boto.swf.layer1, 777
boto.swf.layer1_decisions, 790
boto.swf.layer2, 791
boto.utils, 221
boto.vpc, 801
boto.vpc.customergateway, 817
boto.vpc.dhcpoptions, 817
boto.vpc.internetgateway, 817
boto.vpc.routetable, 818
boto.vpc.subnet, 818
boto.vpc.vpc, 818
boto.vpc.vpnconnection, 820
boto.vpc.vpngateway, 821

A

- ABORT (boto.exception.ResumableTransferDisposition attribute), 219
- ABORT_CUR_PROCESS (boto.exception.ResumableTransferDisposition attribute), 219
- abort_multipart_upload() (boto.glacier.layer1.Layer1 method), 521
- accept_vpc_peering_connection() (boto.vpc.VPCConnection method), 801
- ACCEPTED_STYLES (boto.mturk.question.SelectionAnswer attribute), 610
- access_key (boto.connection.AWSAuthConnection attribute), 214
- access_key (boto.gs.connection.GSConnection attribute), 556
- AccessDeniedException, 495
- AccessLogAttribute (class in boto.ec2.elb.attributes), 506
- AccessToSnapshotDenied, 681
- AccessToSnapshotDeniedFault, 681
- acl (boto.gs.acl.ACL attribute), 540
- ACL (class in boto.gs.acl), 540
- ACL (class in boto.s3.acl), 697
- Action (class in boto.ec2.volumestatus), 462
- ActionSet (class in boto.ec2.volumestatus), 462
- activate_pipeline() (boto.datapipeline.layer1.DataPipelineConnection method), 322
- ActiveCartItem (boto.mws.response.Cart attribute), 622
- ActiveTrustedSigners (class in boto.cloudfront.signers), 241
- activities() (boto.swf.layer2.Domain method), 794
- Activity (class in boto.ec2.autoscale.activity), 182
- ActivityType (class in boto.swf.layer2), 791
- ActivityWorker (class in boto.swf.layer2), 793
- add() (boto.cloudfront.invalidation.InvalidationBatch method), 241
- add() (boto.cloudsearch.document.DocumentServiceConnection method), 261
- add() (boto.cloudsearch2.document.DocumentServiceConnection method), 277
- add() (boto.mturk.layoutparam.LayoutParameters method), 605
- add() (boto.mturk.qualification.Qualifications method), 607
- add_a() (boto.route53.zone.Zone method), 695
- add_alarm_action() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- add_attachments_to_set() (boto.support.layer1.SupportConnection method), 927
- add_attribute() (boto.dynamodb.item.Item method), 344
- add_batch() (boto.dynamodb.batch.BatchList method), 346
- add_batch() (boto.dynamodb.batch.BatchWriteList method), 347
- add_change() (boto.route53.record.ResourceRecordSets method), 693
- add_change_record() (boto.route53.record.ResourceRecordSets method), 694
- add_cname() (boto.route53.zone.Zone method), 695
- add_communication_to_case() (boto.support.layer1.SupportConnection method), 927
- add_cron() (boto.pyami.installers.Installer method), 631
- add_cron() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- add_email_grant() (boto.gs.acl.ACL method), 540
- add_email_grant() (boto.gs.bucket.Bucket method), 541
- add_email_grant() (boto.gs.key.Key method), 560
- add_email_grant() (boto.s3.acl.ACL method), 697
- add_email_grant() (boto.s3.bucket.Bucket method), 698
- add_email_grant() (boto.s3.key.Key method), 716
- add_env() (boto.pyami.installers.Installer method), 631
- add_env() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- add_event_source() (boto.awslambda.layer1.AWSLambdaConnection method), 188
- add_grant() (boto.ec2.securitygroup.IPPermissions

- method), 456
- add_grant() (boto.s3.acl.ACL method), 697
- add_group_email_grant() (boto.gs.acl.ACL method), 540
- add_group_email_grant() (boto.gs.bucket.Bucket method), 541
- add_group_email_grant() (boto.gs.key.Key method), 560
- add_group_grant() (boto.gs.acl.ACL method), 540
- add_group_grant() (boto.gs.key.Key method), 560
- add_init_script() (boto.pyami.installers.Installer method), 631
- add_init_script() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- add_instance_groups() (boto.emr.connection.EmrConnection method), 507
- add_insufficient_data_action() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- add_jobflow_steps() (boto.emr.connection.EmrConnection method), 507
- add_mx() (boto.route53.zone.Zone method), 695
- add_object() (boto.cloudfront.distribution.Distribution method), 235
- add_ok_action() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- add_param() (boto.rds.parametergroup.ParameterGroup method), 654
- add_permission() (boto.sns.SNSConnection method), 749
- add_permission() (boto.sqs.connection.SQSConnection method), 755
- add_permission() (boto.sqs.queue.Queue method), 763
- add_record() (boto.route53.zone.Zone method), 695
- add_role_to_instance_profile() (boto.iam.connection.IAMConnection method), 574
- add_rule() (boto.ec2.securitygroup.SecurityGroup method), 456
- add_rule() (boto.s3.cors.CORSConfiguration method), 714
- add_rule() (boto.s3.lifecycle.Lifecycle method), 730
- add_sdf_from_s3() (boto.cloudsearch.document.DocumentServiceIndex method), 261
- add_sdf_from_s3() (boto.cloudsearch2.document.DocumentServiceIndex method), 277
- add_source_identifier_to_subscription() (boto.rds2.layer1.RDSConnection method), 877
- add_tag() (boto.ec2.ec2object.TaggedEC2Object method), 440
- add_tag() (boto.s3.tagging.TagSet method), 732
- add_tag_set() (boto.s3.tagging.Tags method), 732
- add_tags() (boto.ec2.ec2object.TaggedEC2Object method), 440
- add_tags() (boto.emr.connection.EmrConnection method), 507
- add_tags_to_resource() (boto.rds2.layer1.RDSConnection method), 877
- add_tags_to_stream() (boto.kinesis.layer1.KinesisConnection method), 589
- add_transition() (boto.s3.lifecycle.Transitions method), 731
- add_user_grant() (boto.gs.acl.ACL method), 540
- add_user_grant() (boto.gs.bucket.Bucket method), 541
- add_user_grant() (boto.gs.key.Key method), 560
- add_user_grant() (boto.s3.acl.ACL method), 697
- add_user_grant() (boto.s3.bucket.Bucket method), 698
- add_user_grant() (boto.s3.key.Key method), 717
- add_user_to_group() (boto.iam.connection.IAMConnection method), 574
- add_value() (boto.route53.record.Record method), 693
- add_value() (boto.sdb.item.Item method), 741
- AddInstanceGroupsResponse (class in boto.emr.emrobject), 512
- Address (class in boto.ec2.address), 403
- AdjustmentType (class in boto.ec2.autoscale.policy), 186
- AdultRequirement (class in boto.mturk.qualification), 606
- ALARM (boto.ec2.cloudwatch.alarm.MetricAlarm attribute), 295
- Alarm (class in boto.ec2.autoscale.policy), 186
- AlarmHistoryItem (class in boto.ec2.cloudwatch.alarm), 294
- AliasBody (boto.route53.record.Record attribute), 693
- AllIndex (class in boto.dynamodb2.fields), 349
- allocate_address() (boto.ec2.connection.EC2Connection method), 405
- allow_doc_ip() (boto.cloudsearch.optionstatus.ServicePoliciesStatus method), 257
- allow_doc_ip() (boto.cloudsearch2.optionstatus.ServicePoliciesStatus method), 273
- allow_search_ip() (boto.cloudsearch.optionstatus.ServicePoliciesStatus method), 258
- allow_search_ip() (boto.cloudsearch2.optionstatus.ServicePoliciesStatus method), 273
- ALLOWED_SCOPE_TYPE_SUB_ELEMS (boto.gs.acl.Scope attribute), 540
- AlreadyExistsException (class in boto.mturk.question), 828
- AnswerSpecification (class in boto.mturk.question), 607
- Apache (class in boto.pyami.installers.ubuntu.apache), 632
- APIVersion (boto.awslambda.layer1.AWSLambdaConnection attribute), 188
- APIVersion (boto.beanstalk.layer1.Layer1 attribute), 192
- APIVersion (boto.cloudformation.connection.CloudFormationConnection attribute), 225
- APIVersion (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243
- APIVersion (boto.cloudsearch.layer1.Layer1 attribute),

- 250
- APIVersion (boto.cloudsearch2.layer1.CloudSearchConnection attribute), 266
- APIVersion (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection attribute), 278
- APIVersion (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284
- APIVersion (boto.codedeploy.layer1.CodeDeployConnection attribute), 296
- APIVersion (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307
- APIVersion (boto.cognito.sync.layer1.CognitoSyncConnection attribute), 312
- APIVersion (boto.configservice.layer1.ConfigServiceConnection attribute), 317
- APIVersion (boto.connection.AWSQueryConnection attribute), 215
- APIVersion (boto.datapipeline.layer1.DataPipelineConnection attribute), 322
- APIVersion (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370
- APIVersion (boto.ec2.autoscale.AutoScaleConnection attribute), 176
- APIVersion (boto.ec2.cloudwatch.CloudWatchConnection attribute), 288
- APIVersion (boto.ec2.connection.EC2Connection attribute), 405
- APIVersion (boto.ec2.elb.ELBConnection attribute), 495
- APIVersion (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection attribute), 464
- APIVersion (boto.ecs.ECSCConnection attribute), 470
- APIVersion (boto.elasticache.layer1.ElastiCacheConnection attribute), 471
- APIVersion (boto.elastictranscoder.layer1.ElasticTranscoderConnection attribute), 485
- APIVersion (boto.emr.connection.EmrConnection attribute), 507
- APIVersion (boto.fps.connection.FPSConnection attribute), 518
- APIVersion (boto.iam.connection.IAMConnection attribute), 574
- APIVersion (boto.kinesis.layer1.KinesisConnection attribute), 589
- APIVersion (boto.kms.layer1.KMSConnection attribute), 822
- APIVersion (boto.logs.layer1.CloudWatchLogsConnection attribute), 829
- APIVersion (boto.machinelearning.layer1.MachineLearningConnection attribute), 833
- APIVersion (boto.mturk.connection.MTurkConnection attribute), 601
- APIVersion (boto.opsworks.layer1.OpsWorksConnection attribute), 847
- APIVersion (boto.rds.RDSCConnection attribute), 634
- APIVersion (boto.rds2.layer1.RDSCConnection attribute), 877
- APIVersion (boto.redshift.layer1.RedshiftConnection attribute), 654
- APIVersion (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910
- APIVersion (boto.sdb.connection.SDBConnection attribute), 734
- APIVersion (boto.ses.connection.SESConnection attribute), 745
- APIVersion (boto.sns.SNSConnection attribute), 749
- APIVersion (boto.sqs.connection.SQSConnection attribute), 755
- APIVersion (boto.sts.STSConnection attribute), 770
- APIVersion (boto.support.layer1.SupportConnection attribute), 927
- APNortheast (boto.s3.connection.Location attribute), 710
- app() (boto.sdb.db.key.Key method), 919
- AppCookieStickinessPolicy (class in boto.ec2.elb.policies), 505
- append_field() (boto.mturk.question.OrderedContent method), 609
- Application (class in boto.emr.emrobject), 512
- Application (class in boto.mturk.question), 607
- ApplicationAlreadyExistsException, 303
- ApplicationDescription (class in boto.beanstalk.response), 202
- ApplicationDoesNotExistException, 303
- ApplicationLimitExceededException, 303
- ApplicationNameRequiredException, 303
- ApplicationVersionDescription (class in boto.beanstalk.response), 202
- apply() (boto.rds.parametergroup.Parameter method), 653
- apply_security_groups() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
- apply_security_groups_to_lb() (boto.ec2.elb.ELBConnection method), 496
- approve_assignment() (boto.mturk.connection.MTurkConnection method), 601
- approve_rejected_assignment() (boto.mturk.connection.MTurkConnection method), 601
- APSEast (boto.s3.connection.Location attribute), 711
- APSEast2 (boto.s3.connection.Location attribute), 711
- ArchiveError, 539
- arg (boto.ec2.autoscale.policy.MetricCollectionTypes.BaseType attribute), 186
- arg (boto.ec2.autoscale.policy.MetricCollectionTypes.Granularity attribute), 186
- arg (boto.ec2.autoscale.policy.MetricCollectionTypes.Metric attribute), 186
- Arg (class in boto.emr.emrobject), 513

args() (boto.emr.step.JarStep method), 511
args() (boto.emr.step.Step method), 511
args() (boto.emr.step.StreamingStep method), 512
arn (boto.sqs.queue.Queue attribute), 764
assert_case_insensitive() (in module boto.s3.connection), 714
assign_instance() (boto.opsworks.layer1.OpsWorksConnection method), 847
assign_private_ip_addresses() (boto.ec2.connection.EC2Connection method), 405
assign_qualification() (boto.mturk.connection.MTurkConnection method), 601
assign_volume() (boto.opsworks.layer1.OpsWorksConnection method), 847
Assignment (class in boto.mturk.connection), 600
associate() (boto.ec2.address.Address method), 403
associate_address() (boto.ec2.connection.EC2Connection method), 406
associate_address_object() (boto.ec2.connection.EC2Connection method), 406
associate_dhcp_options() (boto.vpc.VPCCConnection method), 801
associate_elastic_ip() (boto.opsworks.layer1.OpsWorksConnection method), 848
associate_network_acl() (boto.vpc.VPCCConnection method), 801
associate_route_table() (boto.vpc.VPCCConnection method), 802
AssociatedMarketplaces (boto.mws.response.Customer attribute), 623
assume_role() (boto.sts.STSConnection method), 770
assume_role_with_saml() (boto.sts.STSConnection method), 771
assume_role_with_web_identity() (boto.sts.STSConnection method), 772
AssumedRole (class in boto.sts.credentials), 775
attach() (boto.ec2.networkinterface.NetworkInterface method), 452
attach() (boto.ec2.volume.Volume method), 461
attach() (boto.vpc.vpngateway.VpnGateway method), 821
attach_classic_instance() (boto.vpc.vpc.VPC method), 819
attach_classic_link_vpc() (boto.vpc.VPCCConnection method), 802
attach_elastic_load_balancer() (boto.opsworks.layer1.OpsWorksConnection method), 848
attach_group_policy() (boto.iam.connection.IAMConnection method), 574
attach_instances() (boto.ec2.autoscale.AutoScaleConnection method), 176
attach_internet_gateway() (boto.vpc.VPCCConnection method), 802
attach_lb_to_subnets() (boto.ec2.elb.ELBConnection method), 496
attach_network_interface() (boto.ec2.connection.EC2Connection method), 407
attach_role_policy() (boto.iam.connection.IAMConnection method), 574
attach_subnets() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
attach_user_policy() (boto.iam.connection.IAMConnection method), 574
attach_volume() (boto.ec2.connection.EC2Connection method), 407
attach_vpn_gateway() (boto.vpc.VPCCConnection method), 802
Attachment (class in boto.ec2.networkinterface), 452
Attachment (class in boto.vpc.vpngateway), 821
attachment_state() (boto.ec2.volume.Volume method), 461
AttachmentIdNotFound, 931
AttachmentLimitExceeded, 932
AttachmentSet (class in boto.ec2.volume), 460
AttachmentSetExpired, 932
AttachmentSetIdNotFound, 932
AttachmentSetSizeLimitExceeded, 932
attr_type (boto.dynamodb2.fields.BaseSchemaField attribute), 350
attr_type (boto.dynamodb2.fields.HashKey attribute), 352
attr_type (boto.dynamodb2.fields.RangeKey attribute), 353
attribute_names (boto.mturk.question.LengthConstraint attribute), 609
attribute_names (boto.mturk.question.NumericConstraint attribute), 609
attribute_names (boto.mturk.question.RegExConstraint attribute), 610
AttributeList (boto.mws.response.Destination attribute), 623
Attributes (class in boto.sqs.attributes), 755
AttributeSet (class in boto.mws.response), 621
AttributeSets (boto.mws.response.Product attribute), 628
AttrName (boto.ec2.snapshot.Snapshot attribute), 458
auth_region_name (boto.connection.AWSAuthConnection attribute), 214
auth_region_name (boto.gs.connection.GSConnection attribute), 556
auth_service_name (boto.connection.AWSAuthConnection attribute), 214
auth_service_name (boto.gs.connection.GSConnection attribute), 556
AuthorizationAlreadyExists, 681, 874
AuthorizationAlreadyExistsFault, 681

- AuthorizationAmount (boto.mws.response.AuthorizationDetails attribute), 622
- AuthorizationDetails (boto.mws.response.AuthorizeResult attribute), 622
- AuthorizationDetails (class in boto.mws.response), 621
- AuthorizationFee (boto.mws.response.AuthorizationDetails attribute), 622
- AuthorizationNotFound, 681, 874
- AuthorizationNotFoundFault, 681
- AuthorizationQuotaExceeded, 681, 874
- AuthorizationQuotaExceededFault, 681
- AuthorizationStatus (boto.mws.response.AuthorizationDetails attribute), 622
- authorize() (boto.connection.HTTPRequest method), 217
- authorize() (boto.ec2.securitygroup.SecurityGroup method), 456
- authorize() (boto.mws.connection.MWSCConnection method), 611
- authorize() (boto.rds.dbsecuritygroup.DBSecurityGroup method), 651
- authorize_cache_security_group_ingress() (boto.elasticache.layer1.ElastiCacheConnection method), 471
- authorize_cluster_security_group_ingress() (boto.redshift.layer1.RedshiftConnection method), 654
- authorize_db_security_group_ingress() (boto.rds2.layer1.RDSCConnection method), 878
- authorize_dbsecurity_group() (boto.rds.RDSCConnection method), 634
- authorize_security_group() (boto.ec2.connection.EC2Connection method), 407
- authorize_security_group_deprecated() (boto.ec2.connection.EC2Connection method), 408
- authorize_security_group_egress() (boto.ec2.connection.EC2Connection method), 408
- authorize_snapshot_access() (boto.redshift.layer1.RedshiftConnection method), 655
- AuthorizeResult (class in boto.mws.response), 622
- AuthServiceName (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection attribute), 278
- AuthServiceName (boto.machinelearning.layer1.MachineLearningConnection attribute), 833
- AuthServiceName (boto.sqs.connection.SQSConnection attribute), 755
- AuthSMTPHandler (class in boto.utils), 221
- autoscale_build_list_params() (boto.ec2.blockdevicemapping.BlockDeviceMapping method), 404
- AutoScaleConnection (class in boto.ec2.autoscale), 175
- AutoScalingGroup (class in boto.beanstalk.response), 202
- AutoScalingGroup (class in boto.ec2.autoscale.group), 182
- AutoScalingGroupMetric (class in boto.ec2.autoscale.group), 183
- AvailabilityOptionsStatus (class in boto.cloudsearch2.optionstatus), 272
- aws_access_key_id (boto.connection.AWSAuthConnection attribute), 214
- aws_access_key_id (boto.gs.connection.GSConnection attribute), 556
- aws_secret_access_key (boto.connection.AWSAuthConnection attribute), 214
- aws_secret_access_key (boto.gs.connection.GSConnection attribute), 556
- AWSAuthConnection (class in boto.connection), 213
- AWSConnectionError, 218
- AWSLambdaConnection (class in boto.awslambda.layer1), 188
- AWSQueryConnection (class in boto.connection), 215
- ## B
- Backend (class in boto.ec2.elb.loadbalancer), 502
- base64md5 (boto.gs.key.Key attribute), 561
- base64md5 (boto.s3.key.Key attribute), 717
- base_fields (boto.gs.key.Key attribute), 561
- base_fields (boto.s3.key.Key attribute), 717
- base_user_settable_fields (boto.gs.key.Key attribute), 561
- base_user_settable_fields (boto.s3.key.Key attribute), 717
- BaseArgs (boto.emr.step.HiveBase attribute), 510
- BaseArgs (boto.emr.step.PigBase attribute), 511
- BaseAutoResultElement (class in boto.mturk.connection), 600
- BaseIndexField (class in boto.dynamodb2.fields), 349
- BaseObject (class in boto.beanstalk.response), 202
- BaseSchemaField (class in boto.dynamodb2.fields), 349
- Batch (class in boto.dynamodb.batch), 345
- batch_delete_attributes() (boto.sdb.connection.SDBConnection method), 734
- batch_delete_attributes() (boto.sdb.domain.Domain method), 738
- batch_get() (boto.dynamodb2.table.Table method), 358
- batch_get_applications() (boto.codedeploy.layer1.CodeDeployConnection method), 296
- batch_get_deployments() (boto.codedeploy.layer1.CodeDeployConnection method), 297
- batch_get_item() (boto.dynamodb.layer1.Layer1 method), 328
- batch_get_item() (boto.dynamodb.layer2.Layer2 method), 331

batch_get_item() (boto.dynamodb.table.Table method), 338
 batch_get_item() (boto.dynamodb2.layer1.DynamoDBConnection method), 370
 batch_put_attributes() (boto.sdb.connection.SDBConnection method), 734
 batch_put_attributes() (boto.sdb.domain.Domain method), 738
 batch_write() (boto.dynamodb2.table.Table method), 359
 batch_write_item() (boto.dynamodb.layer1.Layer1 method), 328
 batch_write_item() (boto.dynamodb.layer2.Layer2 method), 331
 batch_write_item() (boto.dynamodb2.layer1.DynamoDBConnection method), 371
 BatchGetResultSet (class in boto.dynamodb2.results), 356
 BatchList (class in boto.dynamodb.batch), 346
 BatchResults (class in boto.sqs.batchresults), 768
 BatchTable (class in boto.dynamodb2.table), 357
 BatchWrite (class in boto.dynamodb.batch), 346
 BatchWriteList (class in boto.dynamodb.batch), 347
 BillingBody (boto.gs.bucket.Bucket attribute), 541
 BillingProducts (class in boto.ec2.image), 441
 Binary (class in boto.dynamodb.types), 347
 Binary (class in boto.mturk.question), 607
 Blob (class in boto.sdb.db.blob), 918
 BlobProperty (class in boto.sdb.db.property), 921
 block_worker() (boto.mturk.connection.MTurkConnection method), 601
 BlockDeviceMapping (class in boto.ec2.autoscale.launchconfig), 184
 BlockDeviceMapping (class in boto.ec2.blockdevicemapping), 404
 BlockDeviceType (class in boto.ec2.blockdevicemapping), 404
 BooleanProperty (class in boto.sdb.db.property), 921
 BooleanResult (class in boto.resultset), 220
 Bootstrap (class in boto.pyami.bootstrap), 630
 BootstrapAction (class in boto.emr.emrobject), 513
 BootstrapActionList (class in boto.emr.emrobject), 513
 boto (module), 204
 boto.awslambda (module), 188
 boto.awslambda.exceptions (module), 192
 boto.awslambda.layer1 (module), 188
 boto.beanstalk (module), 192
 boto.beanstalk.layer1 (module), 192
 boto.beanstalk.response (module), 202
 boto.cloudformation (module), 225
 boto.cloudformation.connection (module), 225
 boto.cloudformation.stack (module), 232
 boto.cloudformation.template (module), 234
 boto.cloudfront (module), 234
 boto.cloudfront.distribution (module), 235
 boto.cloudfront.exception (module), 243
 boto.cloudfront.identity (module), 240
 boto.cloudfront.invalidation (module), 241
 boto.cloudfront.logging (module), 243
 boto.cloudfront.object (module), 242
 boto.cloudfront.origin (module), 239
 boto.cloudfront.signers (module), 241
 boto.cloudhsm (module), 243
 boto.cloudhsm.exceptions (module), 246
 boto.cloudhsm.layer1 (module), 243
 boto.cloudsearch (module), 246
 boto.cloudsearch.document (module), 260
 boto.cloudsearch.domain (module), 247
 boto.cloudsearch.layer1 (module), 250
 boto.cloudsearch.layer2 (module), 256
 boto.cloudsearch.optionstatus (module), 257
 boto.cloudsearch.search (module), 258
 boto.cloudsearch2 (module), 262
 boto.cloudsearch2.document (module), 276
 boto.cloudsearch2.domain (module), 262
 boto.cloudsearch2.layer1 (module), 266
 boto.cloudsearch2.layer2 (module), 272
 boto.cloudsearch2.optionstatus (module), 272
 boto.cloudsearch2.search (module), 274
 boto.cloudsearchdomain (module), 278
 boto.cloudsearchdomain.exceptions (module), 283
 boto.cloudsearchdomain.layer1 (module), 278
 boto.cloudtrail (module), 283
 boto.cloudtrail.exceptions (module), 286
 boto.cloudtrail.layer1 (module), 284
 boto.codedeploy (module), 295
 boto.codedeploy.exceptions (module), 303
 boto.codedeploy.layer1 (module), 296
 boto.cognito.identity (module), 306
 boto.cognito.identity.exceptions (module), 311
 boto.cognito.identity.layer1 (module), 307
 boto.cognito.sync (module), 312
 boto.cognito.sync.exceptions (module), 316
 boto.cognito.sync.layer1 (module), 312
 boto.configservice (module), 316
 boto.configservice.exceptions (module), 319
 boto.configservice.layer1 (module), 316
 boto.connection (module), 213
 boto.contrib (module), 321
 boto.contrib.ymlmessage (module), 321
 boto.datapipeline (module), 321
 boto.datapipeline.exceptions (module), 327
 boto.datapipeline.layer1 (module), 321
 boto.dynamodb (module), 327
 boto.dynamodb.batch (module), 345
 boto.dynamodb.item (module), 344
 boto.dynamodb.layer1 (module), 327
 boto.dynamodb.layer2 (module), 331
 boto.dynamodb.schema (module), 343

boto.dynamodb.table (module), 338
boto.dynamodb.types (module), 347
boto.dynamodb2 (module), 369
boto.dynamodb2.exceptions (module), 402
boto.dynamodb2.fields (module), 349
boto.dynamodb2.items (module), 353
boto.dynamodb2.layer1 (module), 369
boto.dynamodb2.results (module), 356
boto.dynamodb2.table (module), 357
boto.ec2 (module), 403
boto.ec2.address (module), 403
boto.ec2.autoscale (module), 175
boto.ec2.autoscale.activity (module), 182
boto.ec2.autoscale.group (module), 182
boto.ec2.autoscale.instance (module), 184
boto.ec2.autoscale.launchconfig (module), 184
boto.ec2.autoscale.policy (module), 186
boto.ec2.autoscale.request (module), 187
boto.ec2.autoscale.scheduled (module), 187
boto.ec2.autoscale.tag (module), 187
boto.ec2.blockdevicemapping (module), 404
boto.ec2.buyreservation (module), 404
boto.ec2.cloudwatch (module), 288
boto.ec2.cloudwatch.alarm (module), 294
boto.ec2.cloudwatch.datapoint (module), 292
boto.ec2.cloudwatch.metric (module), 292
boto.ec2.connection (module), 405
boto.ec2.ec2object (module), 440
boto.ec2.elb (module), 495
boto.ec2.elb.attributes (module), 506
boto.ec2.elb.healthcheck (module), 501
boto.ec2.elb.instancestate (module), 501
boto.ec2.elb.listelement (module), 502
boto.ec2.elb.listener (module), 502
boto.ec2.elb.loadbalancer (module), 502
boto.ec2.elb.policies (module), 505
boto.ec2.elb.securitygroup (module), 505
boto.ec2.group (module), 441
boto.ec2.image (module), 441
boto.ec2.instance (module), 444
boto.ec2.instanceinfo (module), 449
boto.ec2.instancestatus (module), 450
boto.ec2.keypair (module), 451
boto.ec2.launchspecification (module), 451
boto.ec2.networkinterface (module), 452
boto.ec2.placementgroup (module), 454
boto.ec2.regioninfo (module), 454
boto.ec2.reservedinstance (module), 454
boto.ec2.securitygroup (module), 456
boto.ec2.snapshot (module), 457
boto.ec2.spotinstancerequest (module), 458
boto.ec2.tag (module), 460
boto.ec2.volume (module), 460
boto.ec2.volumestatus (module), 462
boto.ec2.zone (module), 463
boto.ec2containerservice (module), 464
boto.ec2containerservice.exceptions (module), 469
boto.ec2containerservice.layer1 (module), 464
boto.ecs (module), 470
boto.ecs.item (module), 470
boto.elasticache (module), 471
boto.elasticache.layer1 (module), 471
boto.elastictranscoder (module), 485
boto.elastictranscoder.exceptions (module), 495
boto.elastictranscoder.layer1 (module), 485
boto.emr (module), 506
boto.emr.connection (module), 506
boto.emr.emrobject (module), 512
boto.emr.step (module), 510
boto.exception (module), 218
boto.file.bucket (module), 515
boto.file.connection (module), 516
boto.file.key (module), 516
boto.file.simpleresultset (module), 516
boto.fps (module), 518
boto.fps.connection (module), 518
boto.glacier (module), 521
boto.glacier.concurrent (module), 538
boto.glacier.exceptions (module), 539
boto.glacier.job (module), 536
boto.glacier.layer1 (module), 521
boto.glacier.layer2 (module), 533
boto.glacier.vault (module), 533
boto.glacier.writer (module), 537
boto.gs.acl (module), 540
boto.gs.bucket (module), 541
boto.gs.bucketlistresultset (module), 555
boto.gs.connection (module), 556
boto.gs.cors (module), 559
boto.gs.key (module), 559
boto.gs.resumable_upload_handler (module), 572
boto.gs.user (module), 572
boto.handler (module), 220
boto.iam (module), 573
boto.iam.connection (module), 574
boto.iam.summarymap (module), 588
boto.kinesis (module), 588
boto.kinesis.exceptions (module), 596
boto.kinesis.layer1 (module), 589
boto.kms (module), 821
boto.kms.exceptions (module), 828
boto.kms.layer1 (module), 822
boto.logs (module), 828
boto.logs.exceptions (module), 833
boto.logs.layer1 (module), 828
boto.machinelearning (module), 833
boto.machinelearning.exceptions (module), 846
boto.machinelearning.layer1 (module), 833

boto.manage (module), 597
boto.manage.cmdshell (module), 597
boto.manage.propget (module), 600
boto.mturk (module), 600
boto.mturk.connection (module), 600
boto.mturk.layoutparam (module), 605
boto.mturk.notification (module), 605
boto.mturk.price (module), 606
boto.mturk.qualification (module), 606
boto.mturk.question (module), 607
boto.mws (module), 611
boto.mws.connection (module), 611
boto.mws.exception (module), 621
boto.mws.response (module), 621
boto.opsworks (module), 846
boto.opsworks.exceptions (module), 874
boto.opsworks.layer1 (module), 847
boto.pyami (module), 630
boto.pyami.bootstrap (module), 630
boto.pyami.config (module), 630
boto.pyami.copybot (module), 631
boto.pyami.installers (module), 631
boto.pyami.installers.ubuntu (module), 632
boto.pyami.installers.ubuntu.apache (module), 632
boto.pyami.installers.ubuntu.installer (module), 632
boto.pyami.installers.ubuntu.mysql (module), 632
boto.pyami.installers.ubuntu.trac (module), 633
boto.pyami.launch_ami (module), 633
boto.pyami.scriptbase (module), 633
boto.pyami.startup (module), 633
boto.rds (module), 634
boto.rds.dbinstance (module), 648
boto.rds.dbsecuritygroup (module), 651
boto.rds.dbsnapshot (module), 652
boto.rds.event (module), 653
boto.rds.parametergroup (module), 653
boto.rds2 (module), 874
boto.rds2.exceptions (module), 874
boto.rds2.layer1 (module), 877
boto.redshift (module), 654
boto.redshift.exceptions (module), 681
boto.redshift.layer1 (module), 654
boto.resultset (module), 220
boto.route53.connection (module), 688
boto.route53.domains (module), 910
boto.route53.domains.exceptions (module), 918
boto.route53.domains.layer1 (module), 910
boto.route53.exception (module), 691
boto.route53.healthcheck (module), 691
boto.route53.hostedzone (module), 693
boto.route53.record (module), 693
boto.route53.status (module), 695
boto.route53.zone (module), 695
boto.s3 (module), 697
boto.s3.acl (module), 697
boto.s3.bucket (module), 698
boto.s3.bucketlistresultset (module), 710
boto.s3.connection (module), 710
boto.s3.cors (module), 714
boto.s3.deletemarker (module), 716
boto.s3.key (module), 716
boto.s3.lifecycle (module), 730
boto.s3.multidelete (module), 728
boto.s3.multipart (module), 727
boto.s3.prefix (module), 727
boto.s3.resumable_download_handler (module), 729
boto.s3.tagging (module), 732
boto.s3.user (module), 732
boto.sdb (module), 733
boto.sdb.connection (module), 733
boto.sdb.db (module), 918
boto.sdb.db.blob (module), 918
boto.sdb.db.key (module), 919
boto.sdb.db.manager (module), 919
boto.sdb.db.manager.sdbmanager (module), 919
boto.sdb.db.property (module), 921
boto.sdb.db.query (module), 925
boto.sdb.domain (module), 738
boto.sdb.item (module), 741
boto.sdb.queryresultset (module), 742
boto.services (module), 743
boto.services.bs (module), 743
boto.services.message (module), 743
boto.services.result (module), 743
boto.services.service (module), 744
boto.services.servicedef (module), 744
boto.services.sonofmmm (module), 744
boto.services.submit (module), 745
boto.ses (module), 745
boto.ses.connection (module), 745
boto.sns (module), 749
boto.sqs (module), 754
boto.sqs.attributes (module), 755
boto.sqs.batchresults (module), 768
boto.sqs.connection (module), 755
boto.sqs.jsonmessage (module), 761
boto.sqs.message (module), 761
boto.sqs.queue (module), 763
boto.sqs.regioninfo (module), 768
boto.sts (module), 769
boto.sts.credentials (module), 775
boto.support (module), 926
boto.support.exceptions (module), 931
boto.support.layer1 (module), 926
boto.swf (module), 777
boto.swf.layer1 (module), 777
boto.swf.layer1_decisions (module), 790
boto.swf.layer2 (module), 791

- boto.utils (module), 221
 - boto.vpc (module), 801
 - boto.vpc.customergateway (module), 817
 - boto.vpc.dhcpoptions (module), 817
 - boto.vpc.internetgateway (module), 817
 - boto.vpc.routetable (module), 818
 - boto.vpc.subnet (module), 818
 - boto.vpc.vpc (module), 818
 - boto.vpc.vpnconnection (module), 820
 - boto.vpc.vpngateway (module), 821
 - BotoClientError, 218
 - BotoServerError, 218
 - BS (class in boto.services.bs), 743
 - Bucket (class in boto.file.bucket), 515
 - Bucket (class in boto.gs.bucket), 541
 - Bucket (class in boto.s3.bucket), 698
 - bucket_lister() (in module boto.s3.bucketlistresultset), 710
 - BucketListResultSet (class in boto.s3.bucketlistresultset), 710
 - BucketNameFilterRequiredException, 303
 - BucketNotFound, 681
 - BucketPaymentBody (boto.s3.bucket.Bucket attribute), 698
 - BUFFER_SIZE (boto.gs.resumable_upload_handler.ResumableUploadHandler attribute), 572
 - BufferSize (boto.gs.key.Key attribute), 560
 - BufferSize (boto.s3.key.Key attribute), 716
 - build_base_http_request() (boto.connection.AWSAuthConnection method), 214
 - build_base_http_request() (boto.gs.connection.GSConnection method), 556
 - build_complex_list_params() (boto.connection.AWSQueryConnection method), 215
 - build_complex_param() (boto.cloudsearch2.layer1.CloudSearchConnection method), 266
 - build_configurations_param_list() (boto.ec2.connection.EC2Connection method), 408
 - build_dimension_param() (boto.ec2.cloudwatch.CloudWatchConnection method), 288
 - build_expects() (boto.dynamodb2.items.Item method), 354
 - build_filter_params() (boto.ec2.connection.EC2Connection method), 408
 - build_key_from_values() (boto.dynamodb.layer2.Layer2 method), 331
 - build_list_params() (boto.connection.AWSQueryConnection method), 216
 - build_list_params() (boto.ec2.autoscale.AutoScaleConnection method), 176
 - build_list_params() (boto.ec2.cloudwatch.CloudWatchConnection method), 288
 - build_list_params() (boto.ec2.elb.ELBConnection method), 496
 - build_list_params() (boto.ec2.networkinterface.NetworkInterfaceCollection method), 453
 - build_params() (boto.ec2.autoscale.tag.Tag method), 188
 - build_path_base() (boto.s3.connection.OrdinaryCallingFormat method), 711
 - build_post_form_args() (boto.gs.connection.GSConnection method), 556
 - build_post_form_args() (boto.s3.connection.S3Connection method), 711
 - build_post_policy() (boto.gs.connection.GSConnection method), 557
 - build_post_policy() (boto.s3.connection.S3Connection method), 712
 - build_put_params() (boto.ec2.cloudwatch.CloudWatchConnection method), 288
 - build_query() (boto.cloudsearch.search.SearchConnection method), 258
 - build_query() (boto.cloudsearch2.search.SearchConnection method), 274
 - build_query() (boto.cloudsearch2.layer1.CloudSearchConnection method), 266
 - build_tag_param_list() (boto.ec2.connection.EC2Connection method), 408
 - build_url_base() (boto.s3.connection.ProtocolIndependentOrdinaryCallingFormat method), 711
 - bundle_instance() (boto.ec2.connection.EC2Connection method), 408
 - Buyer (boto.mws.response.OrderReferenceDetails attribute), 628
 - BuyingPrice (boto.mws.response.Offer attribute), 627
 - BuyReservation (class in boto.ec2.buyreservation), 404
 - ByteTranslatingCallbackHandler (class in boto.s3.resumable_download_handler), 729
- ## C
- calculate_stats() (boto.services.result.ResultProcessor method), 743
 - CalculatedProperty (class in boto.sdb.db.property), 922
 - call() (boto.s3.resumable_download_handler.ByteTranslatingCallbackHandler method), 729
 - cancel() (boto.ec2.spotinstancerequest.SpotInstanceRequest method), 459
 - cancel() (boto.fps.connection.FPSConnection method), 518
 - cancel() (boto.swf.layer2.ActivityWorker method), 793
 - cancel_bundle_task() (boto.ec2.connection.EC2Connection method), 409
 - cancel_feed_submissions() (boto.mws.connection.MWSConnection method), 176

method), 611

cancel_fulfillment_order() (boto.mws.connection.MWSConnection method), 611

cancel_job() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 485

cancel_multipart_upload() (boto.gs.bucket.Bucket method), 542

cancel_multipart_upload() (boto.s3.bucket.Bucket method), 699

cancel_order_reference() (boto.mws.connection.MWSConnection method), 611

cancel_report_requests() (boto.mws.connection.MWSConnection method), 611

cancel_reserved_instances_listing() (boto.ec2.connection.EC2Connection method), 409

cancel_spot_instance_requests() (boto.ec2.connection.EC2Connection method), 409

cancel_subscription_and_refund() (boto.fps.connection.FPSCConnection method), 518

cancel_timer() (boto.swf.layer1_decisions.Layer1Decisions method), 790

cancel_token() (boto.fps.connection.FPSCConnection method), 518

cancel_update_stack() (boto.cloudformation.connection.CloudFormationConnection method), 226

cancel_upload() (boto.s3.multipart.MultiPartUpload method), 727

cancel_workflow_executions() (boto.swf.layer1_decisions.Layer1Decisions method), 790

CancelFeedSubmissionsResult (class in boto.mws.response), 622

CancelReportRequestsResult (class in boto.mws.response), 622

CannedACLStrings (in module boto.gs.acl), 540

canonical_string() (in module boto.utils), 222

Capability (class in boto.cloudformation.stack), 232

capture() (boto.mws.connection.MWSConnection method), 611

CaptureAmount (boto.mws.response.CaptureDetails attribute), 622

CapturedAmount (boto.mws.response.AuthorizationDetails attribute), 622

CaptureDetails (boto.mws.response.CaptureResult attribute), 622

CaptureDetails (class in boto.mws.response), 622

CaptureFee (boto.mws.response.CaptureDetails attribute), 622

CaptureResult (class in boto.mws.response), 622

CaptureStatus (boto.mws.response.CaptureDetails attribute), 622

Cart (class in boto.mws.response), 622

CartItem (class in boto.mws.response), 622

CartsList (boto.mws.response.ListCartsResult attribute), 626

CaseCreationLimitExceeded, 932

CaseIdNotFound, 932

cbui_url() (boto.fps.connection.FPSCConnection method), 518

change_data_dir() (boto.pyami.installers.ubuntu.mysql.MySQL method), 633

change_hit_type_of_hit() (boto.mturk.connection.MTurkConnection method), 601

change_message_visibility() (boto.sqs.connection.SQSConnection method), 755

change_message_visibility_batch() (boto.sqs.connection.SQSConnection method), 756

change_message_visibility_batch() (boto.sqs.queue.Queue method), 764

change_rrsets() (boto.route53.connection.Route53Connection method), 688

change_storage_class() (boto.gs.key.Key method), 561

change_storage_class() (boto.s3.key.Key method), 717

change_visibility() (boto.sqs.message.RawMessage attribute), 760

ChangeResourceRecordSetsBody (boto.route53.record.ResourceRecordSets attribute), 693

ChangeXML (boto.route53.record.ResourceRecordSets attribute), 693

characters() (boto.handler.XmlHandler method), 220

characters() (boto.sdb.domain.DomainDumpParser method), 741

check_dns_availability() (boto.beanstalk.layer1.Layer1 method), 192

check_domain_availability() (boto.route53.domains.layer1.Route53DomainsConnection method), 910

check_instance() (boto.sdb.db.property.ReferenceProperty method), 924

check_lowercase_bucketname() (in module boto.s3.connection), 714

check_uuid() (boto.sdb.db.property.ReferenceProperty method), 925

CheckDNSAvailabilityResponse (class in boto.beanstalk.response), 202

CheckDnsAvailabilityResponse (class in boto.beanstalk.response), 202

clean() (boto.connection.ConnectionPool method), 216

clean() (boto.connection.HostConnectionPool method),

- 217
- CLEAN_INTERVAL (boto.connection.ConnectionPool attribute), 216
- cleanup() (boto.services.service.Service method), 744
- clear() (boto.sqs.queue.Queue method), 764
- clear_sdf() (boto.cloudsearch.document.DocumentServiceConnection method), 261
- clear_sdf() (boto.cloudsearch2.document.DocumentServiceConnection method), 277
- ClientException, 469
- clone_stack() (boto.opsworks.layer1.OpsWorksConnection method), 848
- close() (boto.connection.AWSAuthConnection method), 214
- close() (boto.file.key.Key method), 516
- close() (boto.glacier.writer.Writer method), 537
- close() (boto.gs.connection.GSConnection method), 557
- close() (boto.gs.key.Key method), 561
- close() (boto.manage.cmdshell.LocalClient method), 597
- close() (boto.manage.cmdshell.SSHClient method), 598
- close() (boto.s3.key.Key method), 717
- close_authorization() (boto.mws.connection.MWSConnection method), 612
- close_order_reference() (boto.mws.connection.MWSConnection method), 612
- closed (boto.gs.key.Key attribute), 561
- closed (boto.s3.key.Key attribute), 717
- CloudFormationConnection (class in boto.cloudformation.connection), 225
- CloudFrontConnection (class in boto.cloudfront), 234
- CloudFrontServerError, 243
- CloudHSMConnection (class in boto.cloudhsm.layer1), 243
- CloudHsmInternalException, 246
- CloudHsmServiceException, 246
- CloudSearchConnection (class in boto.cloudsearch2.layer1), 266
- CloudSearchDomainConnection (class in boto.cloudsearchdomain.layer1), 278
- CloudTrailConnection (class in boto.cloudtrail.layer1), 284
- CloudWatchConnection (class in boto.ec2.cloudwatch), 288
- CloudWatchLogsConnection (class in boto.logs.layer1), 828
- CloudWatchLogsDeliveryUnavailableException, 286
- Cluster (class in boto.emr.emrobject), 513
- ClusterAlreadyExists, 681
- ClusterAlreadyExistsFault, 681
- ClusterNotFound, 681
- ClusterNotFoundFault, 681
- ClusterParameterGroupAlreadyExists, 681
- ClusterParameterGroupAlreadyExistsFault, 682
- ClusterParameterGroupNotFound, 682
- ClusterParameterGroupNotFoundFault, 682
- ClusterParameterGroupQuotaExceeded, 682
- ClusterParameterGroupQuotaExceededFault, 682
- ClusterQuotaExceeded, 682
- ClusterQuotaExceededFault, 682
- ClusterSecurityGroupAlreadyExists, 682
- ClusterSecurityGroupAlreadyExistsFault, 682
- ClusterSecurityGroupNotFound, 682
- ClusterSecurityGroupNotFoundFault, 682
- ClusterSecurityGroupQuotaExceeded, 682
- ClusterSecurityGroupQuotaExceededFault, 683
- ClusterSnapshotAlreadyExists, 683
- ClusterSnapshotAlreadyExistsFault, 683
- ClusterSnapshotNotFound, 683
- ClusterSnapshotNotFoundFault, 683
- ClusterSnapshotQuotaExceeded, 683
- ClusterSnapshotQuotaExceededFault, 683
- ClusterStateChangeReason (class in boto.emr.emrobject), 513
- ClusterStatus (class in boto.emr.emrobject), 513
- ClusterSubnetGroupAlreadyExists, 683
- ClusterSubnetGroupAlreadyExistsFault, 683
- ClusterSubnetGroupNotFound, 683
- ClusterSubnetGroupNotFoundFault, 683
- ClusterSubnetGroupQuotaExceeded, 683
- ClusterSubnetGroupQuotaExceededFault, 683
- ClusterSubnetQuotaExceeded, 684
- ClusterSubnetQuotaExceededFault, 684
- ClusterSummary (class in boto.emr.emrobject), 513
- ClusterSummaryList (class in boto.emr.emrobject), 513
- ClusterTimeline (class in boto.emr.emrobject), 513
- CNNorth1 (boto.s3.connection.Location attribute), 711
- CodeDeployConnection (class in boto.codedeploy.layer1), 296
- CODFee (boto.mws.response.OrderItem attribute), 627
- CODFeeDiscount (boto.mws.response.OrderItem attribute), 627
- CognitoIdentityConnection (class in boto.cognito.identity.layer1), 307
- CognitoSyncConnection (class in boto.cognito.sync.layer1), 312
- Commands (boto.services.bs.BS attribute), 743
- commit() (boto.cloudsearch.document.DocumentServiceConnection method), 262
- commit() (boto.cloudsearch2.document.DocumentServiceConnection method), 277
- commit() (boto.route53.record.ResourceRecordSets method), 695
- CommitMismatchError, 258, 260, 276
- CommitResponse (class in boto.cloudsearch.document), 260
- CommitResponse (class in boto.cloudsearch2.document), 276

CompetitivePrice (boto.mws.response.CompetitivePriceList attribute), 622

CompetitivePrice (class in boto.mws.response), 622

CompetitivePriceList (class in boto.mws.response), 622

CompetitivePrices (boto.mws.response.CompetitivePricing attribute), 622

CompetitivePricing (boto.mws.response.Product attribute), 628

CompetitivePricing (class in boto.mws.response), 622

complete() (boto.swf.layer2.ActivityWorker method), 793

complete() (boto.swf.layer2.Decider method), 794

complete_multipart_upload() (boto.glacier.layer1.Layer1 method), 522

complete_multipart_upload() (boto.gs.bucket.Bucket method), 542

complete_multipart_upload() (boto.s3.bucket.Bucket method), 699

complete_upload() (boto.s3.multipart.MultiPartUpload method), 727

complete_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 790

CompleteMultiPartUpload (class in boto.s3.multipart), 727

ComplexAmount (class in boto.mws.response), 623

ComplexDimensions (class in boto.mws.response), 623

ComplexMoney (class in boto.mws.response), 623

ComplexType (class in boto.mws.response), 623

ComplexWeight (class in boto.mws.response), 623

compose() (boto.gs.key.Key method), 561

compute_hash() (boto.gs.key.Key method), 561

compute_hash() (in module boto.utils), 222

compute_md5() (boto.gs.key.Key method), 562

compute_md5() (boto.s3.key.Key method), 717

compute_md5() (in module boto.utils), 223

concurrent_create_archive_from_file() (boto.glacier.vault.Vault method), 533

ConcurrentDownloader (class in boto.glacier.concurrent), 538

ConcurrentTransferer (class in boto.glacier.concurrent), 538

ConcurrentUploader (class in boto.glacier.concurrent), 538

ConditionalCheckFailedError (boto.dynamodb.layer1.Layer1 attribute), 328

ConditionalCheckFailedException, 402

Config (class in boto.pyami.config), 630

ConfigServiceConnection (class in boto.configservice.layer1), 316

ConfigurationOptionDescription (class in boto.beanstalk.response), 202

ConfigurationOptionSetting (class in boto.beanstalk.response), 202

ConfigurationSettingsDescription (class in boto.beanstalk.response), 202

configure_billing() (boto.gs.bucket.Bucket method), 542

configure_health_check() (boto.ec2.elb.ELBConnection method), 496

configure_health_check() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503

configure_lifecycle() (boto.gs.bucket.Bucket method), 542

configure_lifecycle() (boto.s3.bucket.Bucket method), 699

configure_versioning() (boto.gs.bucket.Bucket method), 542

configure_versioning() (boto.s3.bucket.Bucket method), 699

configure_website() (boto.gs.bucket.Bucket method), 542

configure_website() (boto.s3.bucket.Bucket method), 699

confirm_order_reference() (boto.mws.connection.MWSConnection method), 612

confirm_product() (boto.ec2.instance.Instance method), 446

confirm_product_instance() (boto.ec2.connection.EC2Connection method), 409

confirm_subscription() (boto.sns.SNSConnection method), 750

connect() (boto.iam.IAMRegionInfo method), 573

connect() (boto.manage.cmdshell.SSHClient method), 598

connect() (boto.s3.S3RegionInfo method), 697

connect_autoscale() (in module boto), 204

connect_awslambda() (in module boto), 204

connect_beanstalk() (in module boto), 204

connect_cloudformation() (in module boto), 205

connect_cloudfront() (in module boto), 205

connect_cloudhsm() (in module boto), 205

connect_cloudsearch() (in module boto), 205

connect_cloudsearch2() (in module boto), 205

connect_cloudsearchdomain() (in module boto), 205

connect_cloudtrail() (in module boto), 206

connect_cloudwatch() (in module boto), 206

connect_codedeploy() (in module boto), 206

connect_cognito_identity() (in module boto), 206

connect_cognito_sync() (in module boto), 206

connect_configservice() (in module boto), 207

connect_directconnect() (in module boto), 207

connect_dynamodb() (in module boto), 207

connect_ec2() (in module boto), 207

connect_ec2_endpoint() (in module boto), 207

connect_ec2containerservice() (in module boto), 207

connect_elastictranscoder() (in module boto), 208

- connect_elb() (in module boto), 208
- connect_emr() (in module boto), 208
- connect_euca() (in module boto), 208
- connect_fps() (in module boto), 208
- connect_glacier() (in module boto), 208
- connect_gs() (in module boto), 209
- connect_ia() (in module boto), 209
- connect_iam() (in module boto), 209
- connect_kinesis() (in module boto), 209
- connect_kms() (in module boto), 209
- connect_logs() (in module boto), 210
- connect_machinelearning() (in module boto), 210
- connect_mturk() (in module boto), 210
- connect_opsworks() (in module boto), 210
- connect_rds() (in module boto), 210
- connect_rds2() (in module boto), 210
- connect_redshift() (in module boto), 210
- connect_route53() (in module boto), 210
- connect_route53domains() (in module boto), 211
- connect_s3() (in module boto), 211
- connect_sdb() (in module boto), 211
- connect_ses() (in module boto), 211
- connect_sns() (in module boto), 211
- connect_sqs() (in module boto), 211
- connect_sts() (in module boto), 212
- connect_support() (in module boto), 212
- connect_swf() (in module boto), 212
- connect_to_region() (in module boto.awslambda), 188
- connect_to_region() (in module boto.beanstalk), 192
- connect_to_region() (in module boto.cloudformation), 225
- connect_to_region() (in module boto.cloudhsm), 243
- connect_to_region() (in module boto.cloudsearch), 246
- connect_to_region() (in module boto.cloudsearch2), 262
- connect_to_region() (in module boto.cloudsearchdomain), 278
- connect_to_region() (in module boto.cloudtrail), 283
- connect_to_region() (in module boto.codedeploy), 295
- connect_to_region() (in module boto.cognito.identity), 306
- connect_to_region() (in module boto.cognito.sync), 312
- connect_to_region() (in module boto.configservice), 316
- connect_to_region() (in module boto.datapipeline), 321
- connect_to_region() (in module boto.dynamodb), 327
- connect_to_region() (in module boto.dynamodb2), 369
- connect_to_region() (in module boto.ec2), 403
- connect_to_region() (in module boto.ec2.autoscale), 181
- connect_to_region() (in module boto.ec2.cloudwatch), 292
- connect_to_region() (in module boto.ec2.elb), 501
- connect_to_region() (in module boto.ec2.containerservice), 464
- connect_to_region() (in module boto.elasticache), 471
- connect_to_region() (in module boto.elastictranscoder), 485
- connect_to_region() (in module boto.emr), 506
- connect_to_region() (in module boto.glacier), 521
- connect_to_region() (in module boto.iam), 573
- connect_to_region() (in module boto.kinesis), 588
- connect_to_region() (in module boto.kms), 821
- connect_to_region() (in module boto.logs), 828
- connect_to_region() (in module boto.machinelearning), 833
- connect_to_region() (in module boto.opsworks), 846
- connect_to_region() (in module boto.rds), 647
- connect_to_region() (in module boto.rds2), 874
- connect_to_region() (in module boto.redshift), 654
- connect_to_region() (in module boto.route53.domains), 910
- connect_to_region() (in module boto.s3), 697
- connect_to_region() (in module boto.sdb), 733
- connect_to_region() (in module boto.ses), 745
- connect_to_region() (in module boto.sns), 749
- connect_to_region() (in module boto.sqs), 754
- connect_to_region() (in module boto.sts), 769
- connect_to_region() (in module boto.support), 926
- connect_to_region() (in module boto.swf), 777
- connect_to_region() (in module boto.vpc), 817
- connect_vpc() (in module boto), 212
- connect_walrus() (in module boto), 212
- connection (boto.connection.AWSAuthConnection attribute), 214
- connection (boto.gs.connection.GSConnection attribute), 557
- connection (boto.mws.response.ResponseElement attribute), 629
- ConnectionDrainingAttribute (class in boto.ec2.elb.attributes), 506
- ConnectionPool (class in boto.connection), 216
- ConnectionSettingAttribute (class in boto.ec2.elb.attributes), 506
- ConsoleOutput (class in boto.ec2.instance), 444
- ConsoleOutput (class in boto.exception), 218
- Constraint (class in boto.mturk.question), 607
- Constraints (boto.mws.response.OrderReferenceDetails attribute), 628
- Constraints (class in boto.mturk.question), 608
- consumed_units (boto.dynamodb.layer2.TableGenerator attribute), 337
- ContentTooLongError, 261, 276
- continue_as_new_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 790
- convert_binary() (in module boto.dynamodb.types), 348
- convert_num() (in module boto.dynamodb.types), 348
- cooldown (boto.ec2.autoscale.group.AutoScalingGroup attribute), 183

- copy() (boto.gs.key.Key method), 562
- copy() (boto.s3.key.Key method), 718
- copy_bucket_acl() (boto.pyami.copybot.CopyBot method), 631
- copy_cluster_snapshot() (boto.redshift.layer1.RedshiftConnection method), 655
- copy_db_snapshot() (boto.rds2.layer1.RDSConnection method), 878
- copy_dbsnapshot() (boto.rds.RDSConnection method), 634
- copy_image() (boto.ec2.connection.EC2Connection method), 409
- copy_key() (boto.gs.bucket.Bucket method), 542
- copy_key() (boto.s3.bucket.Bucket method), 700
- copy_key_acl() (boto.pyami.copybot.CopyBot method), 631
- copy_keys() (boto.pyami.copybot.CopyBot method), 631
- copy_log() (boto.pyami.copybot.CopyBot method), 631
- copy_part_from_key() (boto.s3.multipart.MultiPartUpload method), 727
- copy_snapshot() (boto.ec2.connection.EC2Connection method), 409
- copy_to_region() (boto.ec2.keypair.KeyPair method), 451
- copy_to_region() (boto.ec2.securitygroup.SecurityGroup method), 457
- CopyBot (class in boto.pyami.copybot), 631
- CopyImage (class in boto.ec2.image), 441
- CopyToRegionDisabled, 684
- Cors (class in boto.gs.cors), 559
- CORSConfiguration (class in boto.s3.cors), 714
- CORSRule (class in boto.s3.cors), 715
- count (boto.dynamodb.layer2.TableGenerator attribute), 337
- count() (boto.dynamodb2.table.Table method), 359
- count() (boto.sdb.db.manager.sdbmanager.SDBManager method), 920
- count() (boto.sdb.db.query.Query method), 925
- count() (boto.sqs.queue.Queue method), 764
- count_closed_workflow_executions() (boto.swf.layer1.Layer1 method), 777
- count_open_workflow_executions() (boto.swf.layer1.Layer1 method), 778
- count_pending_activity_tasks() (boto.swf.layer1.Layer1 method), 778
- count_pending_activity_tasks() (boto.swf.layer2.Domain method), 795
- count_pending_decision_tasks() (boto.swf.layer1.Layer1 method), 778
- count_pending_decision_tasks() (boto.swf.layer2.Domain method), 795
- count_slow() (boto.sqs.queue.Queue method), 764
- create() (boto.dynamodb.schema.Schema class method), 343
- create() (boto.dynamodb2.table.Table class method), 360
- create_access_key() (boto.iam.connection.IAMConnection method), 574
- create_account_alias() (boto.iam.connection.IAMConnection method), 574
- create_alarm() (boto.ec2.cloudwatch.CloudWatchConnection method), 288
- create_alarm() (boto.ec2.cloudwatch.metric.Metric method), 292
- create_alias() (boto.kms.layer1.KMSConnection method), 822
- create_app() (boto.opsworks.layer1.OpsWorksConnection method), 851
- create_app_cookie_stickiness_policy() (boto.ec2.elb.ELBConnection method), 496
- create_app_cookie_stickiness_policy() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
- create_application() (boto.beanstalk.layer1.Layer1 method), 192
- create_application() (boto.codedeploy.layer1.CodeDeployConnection method), 297
- create_application_version() (boto.beanstalk.layer1.Layer1 method), 193
- create_archive_from_file() (boto.glacier.vault.Vault method), 534
- create_archive_writer() (boto.glacier.vault.Vault method), 534
- create_auto_scaling_group() (boto.ec2.autoscale.AutoScaleConnection method), 176
- create_batch_prediction() (boto.machinelearning.layer1.MachineLearningConnection method), 834
- create_bucket() (boto.gs.connection.GSConnection method), 557
- create_bucket() (boto.s3.connection.S3Connection method), 712
- create_cache_cluster() (boto.elasticache.layer1.ElastiCacheConnection method), 471
- create_cache_parameter_group() (boto.elasticache.layer1.ElastiCacheConnection method), 473
- create_cache_security_group() (boto.elasticache.layer1.ElastiCacheConnection method), 474
- create_cache_subnet_group() (boto.elasticache.layer1.ElastiCacheConnection method), 474
- create_case() (boto.support.layer1.SupportConnection method), 927
- create_cluster() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 464
- create_cluster() (boto.redshift.layer1.RedshiftConnection method), 656

create_cluster_parameter_group() (boto.redshift.layer1.RedshiftConnection method), 659
 create_cluster_security_group() (boto.redshift.layer1.RedshiftConnection method), 660
 create_cluster_snapshot() (boto.redshift.layer1.RedshiftConnection method), 660
 create_cluster_subnet_group() (boto.redshift.layer1.RedshiftConnection method), 660
 create_configuration_template() (boto.beanstalk.layer1.Layer1 method), 193
 create_cookie_stickiness_policy() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
 create_customer_gateway() (boto.vpc.VPCConnection method), 803
 create_data_source_from_rds() (boto.machinelearning.layer1.MachineLearningConnection method), 834
 create_data_source_from_redshift() (boto.machinelearning.layer1.MachineLearningConnection method), 835
 create_data_source_from_s3() (boto.machinelearning.layer1.MachineLearningConnection method), 836
 create_db_instance() (boto.rds2.layer1.RDSConnection method), 878
 create_db_instance_read_replica() (boto.rds2.layer1.RDSConnection method), 883
 create_db_parameter_group() (boto.rds2.layer1.RDSConnection method), 884
 create_db_security_group() (boto.rds2.layer1.RDSConnection method), 884
 create_db_snapshot() (boto.rds2.layer1.RDSConnection method), 885
 create_db_subnet_group() (boto.rds.RDSConnection method), 634
 create_db_subnet_group() (boto.rds2.layer1.RDSConnection method), 885
 create_dbinstance() (boto.rds.RDSConnection method), 634
 create_dbinstance_read_replica() (boto.rds.RDSConnection method), 638
 create_dbsecurity_group() (boto.rds.RDSConnection method), 638
 create_dbsnapshot() (boto.rds.RDSConnection method), 638
 create_deployment() (boto.codedeploy.layer1.CodeDeployConnection method), 297
 create_deployment() (boto.opsworks.layer1.OpsWorksConnection method), 852
 create_deployment_config() (boto.codedeploy.layer1.CodeDeployConnection method), 297
 create_deployment_group() (boto.codedeploy.layer1.CodeDeployConnection method), 298
 create_dhcp_options() (boto.vpc.VPCConnection method), 803
 create_distribution() (boto.cloudfront.CloudFrontConnection method), 234
 create_domain() (boto.cloudsearch.layer1.Layer1 method), 250
 create_domain() (boto.cloudsearch.layer2.Layer2 method), 256
 create_domain() (boto.cloudsearch2.layer1.CloudSearchConnection method), 266
 create_domain() (boto.cloudsearch2.layer2.Layer2 method), 272
 create_domain() (boto.sdb.connection.SDBConnection method), 734
 create_environment() (boto.beanstalk.layer1.Layer1 method), 194
 create_evaluation() (boto.machinelearning.layer1.MachineLearningConnection method), 837
 create_event_subscription() (boto.rds2.layer1.RDSConnection method), 885
 create_event_subscription() (boto.redshift.layer1.RedshiftConnection method), 661
 create_expression() (boto.cloudsearch2.domain.Domain method), 263
 create_from_schema() (boto.dynamodb.table.Table class method), 338
 create_fulfillment_order() (boto.mws.connection.MWSConnection method), 612
 create_global_secondary_index() (boto.dynamodb2.table.Table method), 360
 create_grant() (boto.kms.layer1.KMSConnection method), 823
 create_group() (boto.iam.connection.IAMConnection method), 575
 create_hapg() (boto.cloudhsm.layer1.CloudHSMConnection method), 243
 create_health_check() (boto.route53.connection.Route53Connection method), 689
 create_hit() (boto.mturk.connection.MTurkConnection method), 601
 create_hosted_zone() (boto.route53.connection.Route53Connection

method), 689
 create_hsm() (boto.cloudhsm.layer1.CloudHSMConnection method), 243
 create_hsm_client_certificate() (boto.redshift.layer1.RedshiftConnection method), 662
 create_hsm_configuration() (boto.redshift.layer1.RedshiftConnection method), 662
 create_identity_pool() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 307
 create_image() (boto.ec2.connection.EC2Connection method), 410
 create_image() (boto.ec2.instance.Instance method), 446
 create_inbound_shipment() (boto.mws.connection.MWSConnection method), 612
 create_inbound_shipment_plan() (boto.mws.connection.MWSConnection method), 612
 create_index_field() (boto.cloudsearch.domain.Domain method), 247
 create_index_field() (boto.cloudsearch2.domain.Domain method), 264
 create_instance() (boto.opsworks.layer1.OpsWorksConnection method), 852
 create_instance_profile() (boto.iam.connection.IAMConnection method), 575
 create_internet_gateway() (boto.vpc.VPCConnection method), 803
 create_invalidation_request() (boto.cloudfront.CloudFrontConnection method), 234
 create_job() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 486
 create_key() (boto.kms.layer1.KMSConnection method), 823
 create_key_pair() (boto.ec2.connection.EC2Connection method), 410
 create_launch_configuration() (boto.ec2.autoscale.AutoScaleConnection method), 176
 create_layer() (boto.opsworks.layer1.OpsWorksConnection method), 853
 create_lb_cookie_stickiness_policy() (boto.ec2.elb.ELBConnection method), 496
 create_lb_policy() (boto.ec2.elb.ELBConnection method), 497
 create_lb_policy() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
 create_listener() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
 create_listeners() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
 create_load_balancer() (boto.ec2.elb.ELBConnection method), 497
 create_load_balancer_listeners() (boto.ec2.elb.ELBConnection method), 498
 create_log_group() (boto.logs.layer1.CloudWatchLogsConnection method), 829
 create_log_stream() (boto.logs.layer1.CloudWatchLogsConnection method), 829
 create_login_profile() (boto.iam.connection.IAMConnection method), 575
 create_luna_client() (boto.cloudhsm.layer1.CloudHSMConnection method), 244
 create_ml_model() (boto.machinelearning.layer1.MachineLearningConnection method), 838
 create_network_acl() (boto.vpc.VPCConnection method), 804
 create_network_acl_entry() (boto.vpc.VPCConnection method), 804
 create_network_interface() (boto.ec2.connection.EC2Connection method), 410
 create_option_group() (boto.rds.RDSConnection method), 639
 create_option_group() (boto.rds2.layer1.RDSConnection method), 886
 create_or_update_tags() (boto.ec2.autoscale.AutoScaleConnection method), 176
 create_origin_access_identity() (boto.cloudfront.CloudFrontConnection method), 234
 create_parameter_group() (boto.rds.RDSConnection method), 639
 create_pipeline() (boto.datapipeline.layer1.DataPipelineConnection method), 322
 create_pipeline() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 486
 create_placement_group() (boto.ec2.connection.EC2Connection method), 411
 create_platform_application() (boto.sns.SNSConnection method), 750
 create_platform_endpoint() (boto.sns.SNSConnection method), 750
 create_policy() (boto.iam.connection.IAMConnection method), 575
 create_policy_version() (boto.iam.connection.IAMConnection method), 575
 create_preset() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 489
 create_qualification_type() (boto.mturk.connection.MTurkConnection method), 601
 create_queue() (boto.sqs.connection.SQSConnection method), 756

[create_rank_expression\(\)](#) (boto.cloudsearch.domain.Domain method), 248
[create_realtime_endpoint\(\)](#) (boto.machinelearning.layer1.MachineLearningConnection method), 839
[create_replication_group\(\)](#) (boto.elasticache.layer1.ElastiCacheConnection method), 474
[create_reserved_instances_listing\(\)](#) (boto.ec2.connection.EC2Connection method), 411
[create_role\(\)](#) (boto.iam.connection.IAMConnection method), 575
[create_route\(\)](#) (boto.vpc.VPCCConnection method), 804
[create_route_table\(\)](#) (boto.vpc.VPCCConnection method), 805
[create_saml_provider\(\)](#) (boto.iam.connection.IAMConnection method), 575
[create_scaling_policy\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 176
[create_scheduled_group_action\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 176
[create_schema\(\)](#) (boto.dynamodb.layer2.Layer2 method), 332
[create_security_group\(\)](#) (boto.ec2.connection.EC2Connection method), 412
[create_signed_url\(\)](#) (boto.cloudfront.distribution.Distribution method), 236
[create_snapshot\(\)](#) (boto.ec2.connection.EC2Connection method), 412
[create_snapshot\(\)](#) (boto.ec2.volume.Volume method), 461
[create_spot_datafeed_subscription\(\)](#) (boto.ec2.connection.EC2Connection method), 412
[create_stack\(\)](#) (boto.cloudformation.connection.CloudFormationConnection method), 226
[create_stack\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 855
[create_storage_location\(\)](#) (boto.beanstalk.layer1.Layer1 method), 195
[create_stream\(\)](#) (boto.kinesis.layer1.KinesisConnection method), 589
[create_streaming_distribution\(\)](#) (boto.cloudfront.CloudFrontConnection method), 234
[create_subnet\(\)](#) (boto.vpc.VPCCConnection method), 805
[create_subscription\(\)](#) (boto.mws.connection.MWSConnection method), 612
[create_table\(\)](#) (boto.dynamodb.layer1.Layer1 method), 328
[create_table\(\)](#) (boto.dynamodb.layer2.Layer2 method), 332
[create_table\(\)](#) (boto.dynamodb2.layer1.DynamoDBConnection method), 373
[create_tags\(\)](#) (boto.ec2.connection.EC2Connection method), 412
[create_time](#) (boto.dynamodb.table.Table attribute), 339
[create_topic\(\)](#) (boto.sns.SNSConnection method), 751
[create_trail\(\)](#) (boto.cloudtrail.layer1.CloudTrailConnection method), 284
[create_user\(\)](#) (boto.iam.connection.IAMConnection method), 576
[create_user\(\)](#) (boto.pyami.installers.ubuntu.installer.Installer method), 632
[create_user_profile\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 857
[create_vault\(\)](#) (boto.glacier.layer1.Layer1 method), 523
[create_vault\(\)](#) (boto.glacier.layer2.Layer2 method), 533
[create_virtual_mfa_device\(\)](#) (boto.iam.connection.IAMConnection method), 576
[create_volume\(\)](#) (boto.ec2.connection.EC2Connection method), 413
[create_volume\(\)](#) (boto.ec2.snapshot.Snapshot method), 458
[create_vpc\(\)](#) (boto.vpc.VPCCConnection method), 805
[create_vpc_peering_connection\(\)](#) (boto.vpc.VPCCConnection method), 805
[create_vpn_connection\(\)](#) (boto.vpc.VPCCConnection method), 806
[create_vpn_connection_route\(\)](#) (boto.vpc.VPCCConnection method), 806
[create_vpn_gateway\(\)](#) (boto.vpc.VPCCConnection method), 806
[create_working_dir\(\)](#) (boto.pyami.bootstrap.Bootstrap method), 630
[create_zone\(\)](#) (boto.route53.connection.Route53Connection method), 689
[CreateApplicationResponse](#) (class in boto.beanstalk.response), 202
[CreateApplicationVersionResponse](#) (class in boto.beanstalk.response), 202
[CreateConfigurationTemplateResponse](#) (class in boto.beanstalk.response), 203
[created](#) (boto.cloudsearch.domain.Domain attribute), 249
[created](#) (boto.cloudsearch2.domain.Domain attribute), 265
[CreateEnvironmentResponse](#) (class in boto.beanstalk.response), 203
[CreateInboundShipmentPlanResult](#) (class in boto.mws.response), 623
[CreateStorageLocationResponse](#) (class in boto.beanstalk.response), 203
[Credentials](#) (class in boto.sts.credentials), 775
[CrossZoneLoadBalancingAttribute](#) (class in

- boto.ec2.elb.attributes), 506
 - currencycode (boto.fps.connection.FPSConnection attribute), 518
 - current_tree_hash (boto.glacier.writer.Writer attribute), 537
 - current_uploaded_size (boto.glacier.writer.Writer attribute), 537
 - CurrentPrice (boto.mws.response.CartItem attribute), 622
 - Customer (class in boto.mws.response), 623
 - CustomerGateway (class in boto.vpc.customergateway), 817
 - CustomerList (boto.mws.response.ListCustomersResult attribute), 626
 - CustomOrigin (class in boto.cloudfront.origin), 239
- D**
- data_type (boto.sdb.db.property.BlobProperty attribute), 921
 - data_type (boto.sdb.db.property.BooleanProperty attribute), 921
 - data_type (boto.sdb.db.property.DateProperty attribute), 922
 - data_type (boto.sdb.db.property.DateTimeProperty attribute), 922
 - data_type (boto.sdb.db.property.FloatProperty attribute), 922
 - data_type (boto.sdb.db.property.IntegerProperty attribute), 923
 - data_type (boto.sdb.db.property.ListProperty attribute), 923
 - data_type (boto.sdb.db.property.LongProperty attribute), 923
 - data_type (boto.sdb.db.property.MapProperty attribute), 923
 - data_type (boto.sdb.db.property.PasswordProperty attribute), 924
 - data_type (boto.sdb.db.property.Property attribute), 924
 - data_type (boto.sdb.db.property.ReferenceProperty attribute), 925
 - data_type (boto.sdb.db.property.S3KeyProperty attribute), 925
 - data_type (boto.sdb.db.property.TimeProperty attribute), 925
 - DataAlreadyAcceptedException, 833
 - DataPipelineConnection (class in boto.datapipeline.layer1), 321
 - Datapoint (class in boto.ec2.cloudwatch.datapoint), 292
 - date (boto.s3.lifecycle.Transitions attribute), 732
 - DateProperty (class in boto.sdb.db.property), 922
 - DateTimeProperty (class in boto.sdb.db.property), 922
 - days (boto.s3.lifecycle.Transitions attribute), 732
 - DBInstance (class in boto.rds.dbinstance), 648
 - DBInstanceAlreadyExists, 874
 - DBInstanceNotFound, 874
 - DBParameterGroupAlreadyExists, 874
 - DBParameterGroupNotFound, 874
 - DBParameterGroupQuotaExceeded, 874
 - DBSecurityGroup (class in boto.rds.dbsecuritygroup), 651
 - DBSecurityGroupAlreadyExists, 874
 - DBSecurityGroupNotFound, 874
 - DBSecurityGroupNotSupported, 874
 - DBSecurityGroupQuotaExceeded, 875
 - DBSnapshot (class in boto.rds.dbsnapshot), 652
 - DBSnapshotAlreadyExists, 875
 - DBSnapshotNotFound, 875
 - DBSubnetGroupAlreadyExists, 875
 - DBSubnetGroupDoesNotCoverEnoughAZs, 875
 - DBSubnetGroupNotFound, 875
 - DBSubnetGroupQuotaExceeded, 875
 - DBSubnetQuotaExceeded, 875
 - DBUpgradeDependencyFailure, 875
 - deactivate_mfa_device() (boto.iam.connection.IAMConnection method), 576
 - DebuggingArgs (boto.emr.connection.EmrConnection attribute), 507
 - DebuggingJar (boto.emr.connection.EmrConnection attribute), 507
 - Decider (class in boto.swf.layer2), 794
 - DeclarativeType (class in boto.mws.response), 623
 - decode() (boto.dynamodb.types.Dynamizer method), 348
 - decode() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 919
 - decode() (boto.sqs.jsonmessage.JSONMessage method), 761
 - decode() (boto.sqs.message.EncodedMHMessage method), 762
 - decode() (boto.sqs.message.Message method), 763
 - decode() (boto.sqs.message.MHMessage method), 762
 - decode() (boto.sqs.message.RawMessage method), 763
 - decode_authorization_message() (boto.sts.STSConnection method), 773
 - decode_blob() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 919
 - decode_bool() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 919
 - decode_date() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
 - decode_datetime() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
 - decode_float() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
 - decode_int() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
 - decode_list() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
 - decode_long() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920

decode_map() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 192
 method), 920
 DefaultRegionEndpoint (boto.cloudformation.connection.CloudFormationConnection attribute), 225
 decode_map_element() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 225
 method), 920
 DefaultRegionEndpoint (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243
 decode_prop() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 243
 method), 920
 DefaultRegionEndpoint (boto.cloudsearch.layer1.Layer1 attribute), 250
 decode_reference() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 250
 method), 920
 DefaultRegionEndpoint (boto.cloudsearch2.layer1.CloudSearchConnection attribute), 266
 decode_string() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 266
 method), 920
 DefaultRegionEndpoint (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection attribute), 278
 decode_time() (boto.sdb.db.manager.sdbmanager.SDBConverter attribute), 278
 method), 920
 DefaultRegionEndpoint (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284
 decode_value() (boto.sdb.db.manager.sdbmanager.SDBManager attribute), 284
 method), 921
 DefaultRegionEndpoint (boto.codedeploy.layer1.CodeDeployConnection attribute), 296
 decode_value() (boto.sdb.item.Item method), 742
 DecodeAuthorizationMessage (class in boto.sts.credentials), 776
 DefaultRegionEndpoint (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307
 decrypt() (boto.kms.layer1.KMSConnection method), 824
 DefaultRegionEndpoint (boto.cognito.sync.layer1.CognitoSyncConnection attribute), 312
 DEFAULT (boto.gs.connection.Location attribute), 559
 DefaultRegionEndpoint (boto.configservice.layer1.ConfigServiceConnection attribute), 317
 DEFAULT (boto.s3.connection.Location attribute), 711
 default_validator() (boto.sdb.db.property.Property method), 924
 DefaultRegionEndpoint (boto.datapipeline.layer1.DataPipelineConnection attribute), 322
 default_value() (boto.sdb.db.property.DateProperty method), 922
 DefaultRegionEndpoint (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370
 default_value() (boto.sdb.db.property.DateTimeProperty method), 922
 DefaultRegionEndpoint (boto.ec2.autoscale.AutoScaleConnection attribute), 176
 default_value() (boto.sdb.db.property.ListProperty method), 923
 DefaultRegionEndpoint (boto.ec2.cloudwatch.CloudWatchConnection attribute), 288
 default_value() (boto.sdb.db.property.MapProperty method), 923
 DefaultRegionEndpoint (boto.ec2.connection.EC2Connection attribute), 405
 default_value() (boto.sdb.db.property.Property method), 924
 DefaultRegionEndpoint (boto.ec2.elb.ELBConnection attribute), 495
 DefaultCallingFormat (boto.gs.connection.GSConnection attribute), 556
 DefaultRegionEndpoint (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection attribute), 464
 DefaultCallingFormat (boto.s3.connection.S3Connection attribute), 711
 DefaultRegionEndpoint (boto.elasticache.layer1.ElastiCacheConnection attribute), 471
 DefaultContentType (boto.gs.key.Key attribute), 560
 DefaultRegionEndpoint (boto.elastictranscoder.layer1.ElasticTranscoderConnection attribute), 485
 DefaultContentType (boto.s3.key.Key attribute), 716
 DefaultContentType (boto.sqs.connection.SQSConnection attribute), 755
 DefaultRegionEndpoint (boto.emr.connection.EmrConnection attribute), 507
 DefaultHost (boto.cloudfront.CloudFrontConnection attribute), 234
 DefaultRegionEndpoint (boto.kinesis.layer1.KinesisConnection attribute), 589
 DefaultHost (boto.gs.connection.GSConnection attribute), 556
 DefaultRegionEndpoint (boto.kms.layer1.KMSConnection attribute), 822
 DefaultHost (boto.route53.connection.Route53Connection attribute), 688
 DefaultRegionEndpoint (boto.logs.layer1.CloudWatchLogsConnection attribute), 829
 DefaultHost (boto.s3.connection.S3Connection attribute), 711
 DefaultRegionEndpoint (boto.machinelearning.layer1.MachineLearningConnection attribute), 833
 DefaultPartSize (boto.glacier.job.Job attribute), 536
 DefaultRegionEndpoint (boto.opsworks.layer1.OpsWorksConnection attribute), 847
 DefaultPartSize (boto.glacier.vault.Vault attribute), 533
 DefaultRegionEndpoint (boto.awslambda.layer1.AWSLambdaConnection attribute), 188
 DefaultRegionEndpoint (boto.rds.RDSCConnection attribute), 634
 DefaultRegionEndpoint (boto.beanstalk.layer1.Layer1 attribute), 188
 DefaultRegionEndpoint (boto.rds2.layer1.RDSCConnection attribute), 634

attribute), 877

DefaultRegionEndpoint (boto.redshift.layer1.RedshiftConnection attribute), 654

DefaultRegionEndpoint (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910

DefaultRegionEndpoint (boto.sdb.connection.SDBConnection attribute), 734

DefaultRegionEndpoint (boto.ses.connection.SESConnection attribute), 745

DefaultRegionEndpoint (boto.sns.SNSConnection attribute), 749

DefaultRegionEndpoint (boto.sqs.connection.SQSConnection attribute), 755

DefaultRegionEndpoint (boto.sts.STSConnection attribute), 770

DefaultRegionEndpoint (boto.support.layer1.SupportConnection attribute), 927

DefaultRegionName (boto.awslambda.layer1.AWSLambdaConnection attribute), 188

DefaultRegionName (boto.beanstalk.layer1.Layer1 attribute), 192

DefaultRegionName (boto.cloudformation.connection.CloudFormationConnection attribute), 226

DefaultRegionName (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243

DefaultRegionName (boto.cloudsearch.layer1.Layer1 attribute), 250

DefaultRegionName (boto.cloudsearch2.layer1.CloudSearchConnection attribute), 266

DefaultRegionName (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection attribute), 278

DefaultRegionName (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284

DefaultRegionName (boto.codedeploy.layer1.CodeDeployConnection attribute), 296

DefaultRegionName (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307

DefaultRegionName (boto.cognito.sync.layer1.CognitoSyncConnection attribute), 312

DefaultRegionName (boto.configservice.layer1.ConfigServiceConnection attribute), 317

DefaultRegionName (boto.datapipeline.layer1.DataPipelineConnection attribute), 322

DefaultRegionName (boto.dynamodb.layer1.Layer1 attribute), 328

DefaultRegionName (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370

DefaultRegionName (boto.ec2.autoscale.AutoScaleConnection attribute), 176

DefaultRegionName (boto.ec2.cloudwatch.CloudWatchConnection attribute), 288

DefaultRegionName (boto.ec2.connection.EC2Connection attribute), 405

DefaultRegionName (boto.ec2.elb.ELBConnection attribute), 496

DefaultRegionName (boto.ec2containerservice.layer1.EC2ContainerServiceConnection attribute), 464

DefaultRegionName (boto.elasticache.layer1.ElasticacheConnection attribute), 471

DefaultRegionName (boto.elastictranscoder.layer1.ElasticTranscoderConnection attribute), 485

DefaultRegionName (boto.emr.connection.EmrConnection attribute), 507

DefaultRegionName (boto.kinesis.layer1.KinesisConnection attribute), 589

DefaultRegionName (boto.kms.layer1.KMSConnection attribute), 822

DefaultRegionName (boto.logs.layer1.CloudWatchLogsConnection attribute), 829

DefaultRegionName (boto.machinelearning.layer1.MachineLearningConnection attribute), 833

DefaultRegionName (boto.opsworks.layer1.OpsWorksConnection attribute), 847

DefaultRegionName (boto.rds.RDSConnection attribute), 634

DefaultRegionName (boto.rds2.layer1.RDSConnection attribute), 877

DefaultRegionName (boto.redshift.layer1.RedshiftConnection attribute), 654

DefaultRegionName (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910

DefaultRegionName (boto.sdb.connection.SDBConnection attribute), 734

DefaultRegionName (boto.ses.connection.SESConnection attribute), 745

DefaultRegionName (boto.sns.SNSConnection attribute), 749

DefaultRegionName (boto.sqs.connection.SQSConnection attribute), 755

DefaultRegionName (boto.sts.STSConnection attribute), 770

DefaultRegionName (boto.support.layer1.SupportConnection attribute), 927

DefaultRegionName (boto.swf.layer1.Layer1 attribute), 777

analyze_scheme() (boto.cloudsearch2.layer1.CloudSearchConnection method), 267

define_expression() (boto.cloudsearch2.layer1.CloudSearchConnection method), 267

define_index_field() (boto.cloudsearch.layer1.Layer1 method), 250

define_index_field() (boto.cloudsearch2.layer1.CloudSearchConnection method), 267

define_rank_expression() (boto.cloudsearch.layer1.Layer1 method), 251

define_suggester() (boto.cloudsearch2.layer1.CloudSearchConnection method), 267

- method), 267
- definition() (boto.dynamodb2.fields.BaseIndexField method), 349
- definition() (boto.dynamodb2.fields.BaseSchemaField method), 350
- delete() (boto.cloudformation.stack.Stack method), 233
- delete() (boto.cloudfront.distribution.Distribution method), 236
- delete() (boto.cloudfront.distribution.StreamingDistribution method), 238
- delete() (boto.cloudfront.identity.OriginAccessIdentity method), 240
- delete() (boto.cloudsearch.document.DocumentServiceConnection method), 262
- delete() (boto.cloudsearch.domain.Domain method), 249
- delete() (boto.cloudsearch2.document.DocumentServiceConnection method), 277
- delete() (boto.cloudsearch2.domain.Domain method), 265
- delete() (boto.dynamodb.item.Item method), 344
- delete() (boto.dynamodb.table.Table method), 339
- delete() (boto.dynamodb2.items.Item method), 354
- delete() (boto.dynamodb2.table.Table method), 361
- delete() (boto.ec2.address.Address method), 403
- delete() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
- delete() (boto.ec2.autoscale.launchconfig.LaunchConfiguration method), 186
- delete() (boto.ec2.autoscale.policy.ScalingPolicy method), 187
- delete() (boto.ec2.autoscale.tag.Tag method), 188
- delete() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- delete() (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
- delete() (boto.ec2.keypair.KeyPair method), 451
- delete() (boto.ec2.networkinterface.NetworkInterface method), 453
- delete() (boto.ec2.placementgroup.PlacementGroup method), 454
- delete() (boto.ec2.securitygroup.SecurityGroup method), 457
- delete() (boto.ec2.snapshot.Snapshot method), 458
- delete() (boto.ec2.volume.Volume method), 461
- delete() (boto.glacier.vault.Vault method), 534
- delete() (boto.gs.bucket.Bucket method), 543
- delete() (boto.gs.key.Key method), 562
- delete() (boto.rds.dbsecuritygroup.DBSecurityGroup method), 651
- delete() (boto.route53.zone.Zone method), 695
- delete() (boto.s3.bucket.Bucket method), 700
- delete() (boto.s3.key.Key method), 718
- delete() (boto.sdb.domain.Domain method), 739
- delete() (boto.sdb.item.Item method), 742
- delete() (boto.sqs.message.RawMessage method), 763
- delete() (boto.sqs.queue.Queue method), 764
- delete() (boto.vpc.vpc.VPC method), 819
- delete() (boto.vpc.vpnconnection.VpnConnection method), 820
- delete_a() (boto.route53.zone.Zone method), 695
- delete_access_key() (boto.iam.connection.IAMConnection method), 576
- delete_account_alias() (boto.iam.connection.IAMConnection method), 576
- delete_account_password_policy() (boto.iam.connection.IAMConnection method), 577
- delete_alarms() (boto.ec2.cloudwatch.CloudWatchConnection method), 289
- delete_alias() (boto.kms.layer1.KMSConnection method), 824
- delete_analysis_scheme() (boto.cloudsearch2.layer1.CloudSearchConnection method), 268
- delete_app() (boto.opsworks.layer1.OpsWorksConnection method), 857
- delete_application() (boto.beanstalk.layer1.Layer1 method), 195
- delete_application() (boto.codedeploy.layer1.CodeDeployConnection method), 298
- delete_application_version() (boto.beanstalk.layer1.Layer1 method), 195
- delete_archive() (boto.glacier.layer1.Layer1 method), 523
- delete_archive() (boto.glacier.vault.Vault method), 534
- delete_attribute() (boto.dynamodb.item.Item method), 344
- delete_attributes() (boto.sdb.connection.SDBConnection method), 735
- delete_attributes() (boto.sdb.domain.Domain method), 739
- delete_auto_scaling_group() (boto.ec2.autoscale.AutoScaleConnection method), 177
- delete_batch_prediction() (boto.machinelearning.layer1.MachineLearningConnection method), 839
- delete_bucket() (boto.gs.connection.GSConnection method), 557
- delete_bucket() (boto.s3.connection.S3Connection method), 712
- delete_cache_cluster() (boto.elasticache.layer1.ElastiCacheConnection method), 474
- delete_cache_parameter_group() (boto.elasticache.layer1.ElastiCacheConnection method), 475
- delete_cache_security_group() (boto.elasticache.layer1.ElastiCacheConnection method), 475

delete_cache_subnet_group() (boto.elasticache.layer1.ElastiCacheConnection method), 475
 delete_cluster() (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection method), 464
 delete_cluster() (boto.redshift.layer1.RedshiftConnection method), 662
 delete_cluster_parameter_group() (boto.redshift.layer1.RedshiftConnection method), 663
 delete_cluster_security_group() (boto.redshift.layer1.RedshiftConnection method), 663
 delete_cluster_snapshot() (boto.redshift.layer1.RedshiftConnection method), 663
 delete_cluster_subnet_group() (boto.redshift.layer1.RedshiftConnection method), 663
 delete_cname() (boto.route53.zone.Zone method), 695
 delete_configuration_template() (boto.beanstalk.layer1.Layer1 method), 195
 delete_cors() (boto.gs.bucket.Bucket method), 543
 delete_cors() (boto.s3.bucket.Bucket method), 700
 delete_customer_gateway() (boto.vpc.VPCConnection method), 806
 delete_data_source() (boto.machinelearning.layer1.MachineLearningConnection method), 839
 delete_dataset() (boto.cognito.sync.layer1.CognitoSyncConnection method), 312
 delete_db_instance() (boto.rds2.layer1.RDSConnection method), 887
 delete_db_parameter_group() (boto.rds2.layer1.RDSConnection method), 887
 delete_db_security_group() (boto.rds2.layer1.RDSConnection method), 888
 delete_db_snapshot() (boto.rds2.layer1.RDSConnection method), 888
 delete_db_subnet_group() (boto.rds.RDSConnection method), 639
 delete_db_subnet_group() (boto.rds2.layer1.RDSConnection method), 888
 delete_dbinstance() (boto.rds.RDSConnection method), 639
 delete_dbsecurity_group() (boto.rds.RDSConnection method), 639
 delete_dbsnapshot() (boto.rds.RDSConnection method), 639
 delete_delivery_channel() (boto.configservice.layer1.ConfigServiceConnection method), 317
 delete_deployment_config() (boto.codedeploy.layer1.CodeDeployConnection method), 299
 delete_elastic_load_balancing_group() (boto.codedeploy.layer1.CodeDeployConnection method), 299
 delete_dhcp_options() (boto.vpc.VPCConnection method), 807
 delete_distribution() (boto.cloudfront.CloudFrontConnection method), 234
 delete_domain() (boto.cloudsearch.layer1.Layer1 method), 252
 delete_domain() (boto.cloudsearch2.layer1.CloudSearchConnection method), 268
 delete_domain() (boto.sdb.connection.SDBConnection method), 735
 delete_endpoint() (boto.sns.SNSConnection method), 751
 delete_environment_configuration() (boto.beanstalk.layer1.Layer1 method), 196
 delete_evaluation() (boto.machinelearning.layer1.MachineLearningConnection method), 839
 delete_event_subscription() (boto.rds2.layer1.RDSConnection method), 888
 delete_event_subscription() (boto.redshift.layer1.RedshiftConnection method), 663
 delete_expression() (boto.cloudsearch2.layer1.CloudSearchConnection method), 268
 delete_function() (boto.awslambda.layer1.AWSLambdaConnection method), 189
 delete_global_secondary_index() (boto.dynamodb2.table.Table method), 361
 delete_group() (boto.iam.connection.IAMConnection method), 577
 delete_group_policy() (boto.iam.connection.IAMConnection method), 577
 delete_hapg() (boto.cloudhsm.layer1.CloudHSMConnection method), 244
 delete_health_check() (boto.route53.connection.Route53Connection method), 689
 delete_hosted_zone() (boto.route53.connection.Route53Connection method), 690
 delete_hsm() (boto.cloudhsm.layer1.CloudHSMConnection method), 244
 delete_hsm_client_certificate() (boto.redshift.layer1.RedshiftConnection method), 664
 delete_hsm_configuration() (boto.redshift.layer1.RedshiftConnection method), 664
 delete_identity() (boto.ses.connection.SESConnection method), 745

[delete_identity_pool\(\)](#) (boto.cognito.identity.layer1.CognitoIdentityConnection method), 307
[delete_index_field\(\)](#) (boto.cloudsearch.layer1.Layer1 method), 252
[delete_index_field\(\)](#) (boto.cloudsearch2.layer1.CloudSearchConnection method), 268
[delete_instance\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 858
[delete_instance_profile\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_internet_gateway\(\)](#) (boto.vpc.VPCConnection method), 807
[delete_item\(\)](#) (boto.dynamodb.layer1.Layer1 method), 328
[delete_item\(\)](#) (boto.dynamodb.layer2.Layer2 method), 332
[delete_item\(\)](#) (boto.dynamodb2.layer1.DynamoDBConnection method), 375
[delete_item\(\)](#) (boto.dynamodb2.table.BatchTable method), 357
[delete_item\(\)](#) (boto.dynamodb2.table.Table method), 361
[delete_item\(\)](#) (boto.sdb.domain.Domain method), 739
[delete_key\(\)](#) (boto.file.bucket.Bucket method), 515
[delete_key\(\)](#) (boto.gs.bucket.Bucket method), 543
[delete_key\(\)](#) (boto.s3.bucket.Bucket method), 700
[delete_key_pair\(\)](#) (boto.ec2.connection.EC2Connection method), 413
[delete_key_value\(\)](#) (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
[delete_keys\(\)](#) (boto.gs.bucket.Bucket method), 543
[delete_keys\(\)](#) (boto.s3.bucket.Bucket method), 701
[delete_launch_configuration\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 177
[delete_layer\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 858
[delete_lb_policy\(\)](#) (boto.ec2.elb.ELBConnection method), 498
[delete_lifecycle_configuration\(\)](#) (boto.gs.bucket.Bucket method), 544
[delete_lifecycle_configuration\(\)](#) (boto.s3.bucket.Bucket method), 701
[delete_listener\(\)](#) (boto.ec2.elb.loadbalancer.LoadBalancer method), 503
[delete_listeners\(\)](#) (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
[delete_load_balancer\(\)](#) (boto.ec2.elb.ELBConnection method), 498
[delete_load_balancer_listeners\(\)](#) (boto.ec2.elb.ELBConnection method), 498
[delete_log_group\(\)](#) (boto.logs.layer1.CloudWatchLogsConnection method), 830
[delete_log_stream\(\)](#) (boto.logs.layer1.CloudWatchLogsConnection method), 830
[delete_managed_policy\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_luna_client\(\)](#) (boto.cloudhsm.layer1.CloudHSMConnection method), 244
[delete_message\(\)](#) (boto.services.service.Service method), 744
[delete_message\(\)](#) (boto.sqs.connection.SQSConnection method), 756
[delete_message\(\)](#) (boto.sqs.queue.Queue method), 764
[delete_message_batch\(\)](#) (boto.sqs.connection.SQSConnection method), 756
[delete_message_batch\(\)](#) (boto.sqs.queue.Queue method), 764
[delete_message_from_handle\(\)](#) (boto.sqs.connection.SQSConnection method), 757
[delete_metric_filter\(\)](#) (boto.logs.layer1.CloudWatchLogsConnection method), 830
[delete_ml_model\(\)](#) (boto.machinelearning.layer1.MachineLearningConnection method), 839
[delete_mx\(\)](#) (boto.route53.zone.Zone method), 695
[delete_network_acl\(\)](#) (boto.vpc.VPCConnection method), 807
[delete_network_acl_entry\(\)](#) (boto.vpc.VPCConnection method), 807
[delete_network_interface\(\)](#) (boto.ec2.connection.EC2Connection method), 413
[delete_notification_configuration\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 177
[delete_notification_configuration\(\)](#) (boto.ec2.autoscale.group.AutoScalingGroup method), 183
[delete_object\(\)](#) (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
[delete_option_group\(\)](#) (boto.rds.RDSCConnection method), 640
[delete_option_group\(\)](#) (boto.rds2.layer1.RDSCConnection method), 888
[delete_origin_access_identity\(\)](#) (boto.cloudfront.CloudFrontConnection method), 234
[delete_parameter_group\(\)](#) (boto.rds.RDSCConnection method), 640
[delete_pipeline\(\)](#) (boto.datapipeline.layer1.DataPipelineConnection method), 323
[delete_pipeline\(\)](#) (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 489
[delete_placement_group\(\)](#) (boto.ec2.connection.EC2Connection method), 413
[delete_platform_application\(\)](#) (boto.sns.SNSConnection method), 751

[delete_policy\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 177
[delete_policy\(\)](#) (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
[delete_policy\(\)](#) (boto.gs.bucket.Bucket method), 544
[delete_policy\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_policy\(\)](#) (boto.s3.bucket.Bucket method), 701
[delete_policy_version\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_preset\(\)](#) (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 490
[delete_queue\(\)](#) (boto.sqs.connection.SQSConnection method), 757
[delete_rank_expression\(\)](#) (boto.cloudsearch.layer1.Layer1 method), 253
[delete_realtime_endpoint\(\)](#) (boto.machinelearning.layer1.MachineLearningConnection method), 840
[delete_record\(\)](#) (boto.route53.zone.Zone method), 696
[delete_replication_group\(\)](#) (boto.elasticache.layer1.ElastiCacheConnection method), 475
[delete_retention_policy\(\)](#) (boto.logs.layer1.CloudWatchLogsConnection method), 830
[delete_role\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_role_policy\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_route\(\)](#) (boto.vpc.VPCCConnection method), 807
[delete_route_table\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_saml_provider\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_scheduled_action\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 177
[delete_security_group\(\)](#) (boto.ec2.connection.EC2Connection method), 413
[delete_server_cert\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_signing_cert\(\)](#) (boto.iam.connection.IAMConnection method), 577
[delete_snapshot\(\)](#) (boto.ec2.connection.EC2Connection method), 414
[delete_spot_datafeed_subscription\(\)](#) (boto.ec2.connection.EC2Connection method), 414
[delete_stack\(\)](#) (boto.cloudformation.connection.CloudFormationConnection method), 227
[delete_stack\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 858
[delete_stream\(\)](#) (boto.kinesis.layer1.KinesisConnection method), 590
[delete_streaming_distribution\(\)](#) (boto.cloudfront.CloudFrontConnection method), 234
[delete_subnet\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_subscription\(\)](#) (boto.mws.connection.MWSConnection method), 612
[delete_suggester\(\)](#) (boto.cloudsearch2.layer1.CloudSearchConnection method), 268
[delete_table\(\)](#) (boto.dynamodb.layer1.Layer1 method), 329
[delete_table\(\)](#) (boto.dynamodb.layer2.Layer2 method), 333
[delete_table\(\)](#) (boto.dynamodb2.layer1.DynamoDBConnection method), 379
[delete_tags\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 178
[delete_tags\(\)](#) (boto.ec2.connection.EC2Connection method), 414
[delete_tags\(\)](#) (boto.gs.bucket.Bucket method), 544
[delete_tags\(\)](#) (boto.s3.bucket.Bucket method), 701
[delete_topic\(\)](#) (boto.sns.SNSConnection method), 751
[delete_trail\(\)](#) (boto.cloudtrail.layer1.CloudTrailConnection method), 284
[delete_user\(\)](#) (boto.iam.connection.IAMConnection method), 578
[delete_user_policy\(\)](#) (boto.iam.connection.IAMConnection method), 578
[delete_user_profile\(\)](#) (boto.opsworks.layer1.OpsWorksConnection method), 858
[delete_vault\(\)](#) (boto.glacier.layer1.Layer1 method), 523
[delete_vault\(\)](#) (boto.glacier.layer2.Layer2 method), 533
[delete_vault_notifications\(\)](#) (boto.glacier.layer1.Layer1 method), 524
[delete_verified_email_address\(\)](#) (boto.ses.connection.SESConnection method), 745
[delete_volume\(\)](#) (boto.ec2.connection.EC2Connection method), 414
[delete_vpc\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_vpc_peering_connection\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_vpn_connection\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_vpn_connection_route\(\)](#) (boto.vpc.VPCCConnection method), 808
[delete_vpn_gateway\(\)](#) (boto.vpc.VPCCConnection method), 809
[delete_website_configuration\(\)](#) (boto.gs.bucket.Bucket method), 544
[delete_website_configuration\(\)](#) (boto.s3.bucket.Bucket method), 701
[DeleteApplicationResponse](#) (class in

boto.beanstalk.response), 203

DeleteApplicationVersionResponse (class in boto.beanstalk.response), 203

DeleteConfigurationTemplateResponse (class in boto.beanstalk.response), 203

deleted (boto.cloudsearch.domain.Domain attribute), 249

deleted (boto.cloudsearch2.domain.Domain attribute), 265

Deleted (class in boto.s3.multidelete), 728

DeleteEnvironmentConfigurationResponse (class in boto.beanstalk.response), 203

DeleteMarker (class in boto.s3.deletemarker), 716

deliver_config_snapshot() (boto.configservice.layer1.ConfigServiceConnection method), 317

DependencyTimeoutException, 828

DeploymentAlreadyCompletedException, 303

DeploymentConfigAlreadyExistsException, 303

DeploymentConfigDoesNotExistException, 303

DeploymentConfigInUseException, 303

DeploymentConfigLimitExceededException, 303

DeploymentConfigNameRequiredException, 303

DeploymentDoesNotExistException, 304

DeploymentGroupAlreadyExistsException, 304

DeploymentGroupDoesNotExistException, 304

DeploymentGroupLimitExceededException, 304

DeploymentGroupNameRequiredException, 304

DeploymentIdRequiredException, 304

DeploymentLimitExceededException, 304

DeploymentNotStartedException, 304

deprecate() (boto.swf.layer2.ActivityType method), 791

deprecate() (boto.swf.layer2.Domain method), 795

deprecate() (boto.swf.layer2.WorkflowType method), 799

deprecate_activity_type() (boto.swf.layer1.Layer1 method), 779

deprecate_domain() (boto.swf.layer1.Layer1 method), 779

deprecate_workflow_type() (boto.swf.layer1.Layer1 method), 779

deregister() (boto.ec2.image.Image method), 441

deregister_container_instance() (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection method), 464

deregister_destination() (boto.mws.connection.MWSConnection method), 612

deregister_elastic_ip() (boto.opsworks.layer1.OpsWorksConnection method), 858

deregister_image() (boto.ec2.connection.EC2Connection method), 414

deregister_instance() (boto.opsworks.layer1.OpsWorksConnection method), 858

deregister_instances() (boto.ec2.elb.ELBConnection method), 498

deregister_instances() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504

deregister_rds_db_instance() (boto.opsworks.layer1.OpsWorksConnection method), 859

deregister_task_definition() (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection method), 465

deregister_volume() (boto.opsworks.layer1.OpsWorksConnection method), 859

describe() (boto.dynamodb2.table.Table method), 362

describe() (boto.ec2.reservedinstance.ReservedInstancesOffering method), 456

describe() (boto.swf.layer2.ActivityType method), 792

describe() (boto.swf.layer2.Domain method), 796

describe() (boto.swf.layer2.WorkflowExecution method), 797

describe() (boto.swf.layer2.WorkflowType method), 799

describe_account_attributes() (boto.ec2.connection.EC2Connection method), 414

describe_activity_type() (boto.swf.layer1.Layer1 method), 779

describe_alarm_history() (boto.ec2.cloudwatch.CloudWatchConnection method), 289

describe_alarms() (boto.ec2.cloudwatch.CloudWatchConnection method), 289

describe_alarms() (boto.ec2.cloudwatch.metric.Metric method), 293

describe_alarms_for_metric() (boto.ec2.cloudwatch.CloudWatchConnection method), 289

describe_analysis_schemes() (boto.cloudsearch2.layer1.CloudSearchConnection method), 269

describe_application_versions() (boto.beanstalk.layer1.Layer1 method), 196

describe_applications() (boto.beanstalk.layer1.Layer1 method), 196

describe_apps() (boto.opsworks.layer1.OpsWorksConnection method), 859

describe_attachment() (boto.support.layer1.SupportConnection method), 928

describe_availability_options() (boto.cloudsearch2.layer1.CloudSearchConnection method), 269

describe_batch_predictions() (boto.machinelearning.layer1.MachineLearningConnection method), 840

describe_cache_clusters() (boto.elasticache.layer1.ElastiCacheConnection method), 475

describe_cache_engine_versions() (boto.elasticache.layer1.ElastiCacheConnection method), 475

method), 476

describe_cache_parameter_groups()
(boto.elasticache.layer1.ElastiCacheConnection method), 476

describe_cache_parameters()
(boto.elasticache.layer1.ElastiCacheConnection method), 477

describe_cache_security_groups()
(boto.elasticache.layer1.ElastiCacheConnection method), 477

describe_cache_subnet_groups()
(boto.elasticache.layer1.ElastiCacheConnection method), 477

describe_cases() (boto.support.layer1.SupportConnection method), 929

describe_cluster() (boto.emr.connection.EmrConnection method), 507

describe_cluster_parameter_groups()
(boto.redshift.layer1.RedshiftConnection method), 664

describe_cluster_parameters()
(boto.redshift.layer1.RedshiftConnection method), 664

describe_cluster_security_groups()
(boto.redshift.layer1.RedshiftConnection method), 665

describe_cluster_snapshots()
(boto.redshift.layer1.RedshiftConnection method), 665

describe_cluster_subnet_groups()
(boto.redshift.layer1.RedshiftConnection method), 666

describe_cluster_versions()
(boto.redshift.layer1.RedshiftConnection method), 667

describe_clusters() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 465

describe_clusters() (boto.redshift.layer1.RedshiftConnection method), 667

describe_commands() (boto.opsworks.layer1.OpsWorksConnection method), 859

describe_communications()
(boto.support.layer1.SupportConnection method), 929

describe_configuration_options()
(boto.beanstalk.layer1.Layer1 method), 196

describe_configuration_recorder_status()
(boto.configservice.layer1.ConfigServiceConnection method), 317

describe_configuration_recorders()
(boto.configservice.layer1.ConfigServiceConnection method), 317

describe_configuration_settings()
(boto.beanstalk.layer1.Layer1 method), 197

describe_container_instances()
(boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 465

describe_data_sources() (boto.machinelearning.layer1.MachineLearningConnection method), 841

describe_dataset() (boto.cognito.sync.layer1.CognitoSyncConnection method), 312

describe_db_engine_versions()
(boto.rds2.layer1.RDSCConnection method), 888

describe_db_instances() (boto.rds2.layer1.RDSCConnection method), 889

describe_db_log_files() (boto.rds2.layer1.RDSCConnection method), 889

describe_db_parameter_groups()
(boto.rds2.layer1.RDSCConnection method), 890

describe_db_parameters()
(boto.rds2.layer1.RDSCConnection method), 890

describe_db_security_groups()
(boto.rds2.layer1.RDSCConnection method), 891

describe_db_snapshots() (boto.rds2.layer1.RDSCConnection method), 891

describe_db_subnet_groups()
(boto.rds2.layer1.RDSCConnection method), 892

describe_default_cluster_parameters()
(boto.redshift.layer1.RedshiftConnection method), 667

describe_default_search_field()
(boto.cloudsearch.layer1.Layer1 method), 253

describe_delivery_channel_status()
(boto.configservice.layer1.ConfigServiceConnection method), 317

describe_delivery_channels()
(boto.configservice.layer1.ConfigServiceConnection method), 318

describe_deployments() (boto.opsworks.layer1.OpsWorksConnection method), 860

describe_domain() (boto.swf.layer1.Layer1 method), 779

describe_domains() (boto.cloudsearch.layer1.Layer1 method), 253

describe_domains() (boto.cloudsearch2.layer1.CloudSearchConnection method), 269

describe_elastic_ips() (boto.opsworks.layer1.OpsWorksConnection method), 860

describe_elastic_load_balancers()
(boto.opsworks.layer1.OpsWorksConnection method), 860

describe_engine_default_parameters()
(boto.elasticache.layer1.ElastiCacheConnection

- method), 478
- describe_engine_default_parameters()
(boto.rds2.layer1.RDSConnection method),
892
- describe_environment_resources()
(boto.beanstalk.layer1.Layer1 method), 197
- describe_environments() (boto.beanstalk.layer1.Layer1
method), 197
- describe_evaluations() (boto.machinelearning.layer1.MachineLearningConnection method), 842
- describe_event_categories()
(boto.rds2.layer1.RDSConnection method),
893
- describe_event_categories()
(boto.redshift.layer1.RedshiftConnection
method), 668
- describe_event_subscriptions()
(boto.rds2.layer1.RDSConnection method),
893
- describe_event_subscriptions()
(boto.redshift.layer1.RedshiftConnection
method), 668
- describe_events() (boto.beanstalk.layer1.Layer1 method),
198
- describe_events() (boto.cloudformation.stack.Stack
method), 233
- describe_events() (boto.elasticache.layer1.ElastiCacheConnection
method), 478
- describe_events() (boto.rds2.layer1.RDSConnection
method), 893
- describe_events() (boto.redshift.layer1.RedshiftConnection
method), 668
- describe_expressions() (boto.cloudsearch2.layer1.CloudSearchConnection
method), 269
- describe_hapg() (boto.cloudhsm.layer1.CloudHSMConnection
method), 244
- describe_history() (boto.ec2.cloudwatch.alarm.MetricAlarmConnection
method), 295
- describe_hsm() (boto.cloudhsm.layer1.CloudHSMConnection
method), 244
- describe_hsm_client_certificates()
(boto.redshift.layer1.RedshiftConnection
method), 669
- describe_hsm_configurations()
(boto.redshift.layer1.RedshiftConnection
method), 670
- describe_identity_pool() (boto.cognito.identity.layer1.CognitoIdentityConnection
method), 308
- describe_identity_pool_usage()
(boto.cognito.sync.layer1.CognitoSyncConnection
method), 313
- describe_identity_usage()
(boto.cognito.sync.layer1.CognitoSyncConnection
method), 313
- describe_index_fields() (boto.cloudsearch.layer1.Layer1
method), 253
- describe_index_fields() (boto.cloudsearch2.layer1.CloudSearchConnection
method), 270
- describe_instance_health() (boto.ec2.elb.ELBConnection
method), 498
- describe_instances() (boto.opsworks.layer1.OpsWorksConnection
method), 861
- describe_instances() (boto.glacier.layer1.Layer1 method), 524
- describe_jobflow() (boto.emr.connection.EmrConnection
method), 507
- describe_jobflows() (boto.emr.connection.EmrConnection
method), 507
- describe_key() (boto.kms.layer1.KMSConnection
method), 824
- describe_layers() (boto.opsworks.layer1.OpsWorksConnection
method), 861
- describe_load_based_auto_scaling()
(boto.opsworks.layer1.OpsWorksConnection
method), 861
- describe_log_groups() (boto.logs.layer1.CloudWatchLogsConnection
method), 830
- describe_log_streams() (boto.logs.layer1.CloudWatchLogsConnection
method), 830
- describe_logging_status()
(boto.redshift.layer1.RedshiftConnection
method), 670
- describe_luna_client() (boto.cloudhsm.layer1.CloudHSMConnection
method), 244
- describe_metric_filters() (boto.logs.layer1.CloudWatchLogsConnection
method), 831
- describe_ml_models() (boto.machinelearning.layer1.MachineLearningConnection
method), 843
- describe_my_user_profile()
(boto.opsworks.layer1.OpsWorksConnection
method), 861
- describe_objects() (boto.datapipeline.layer1.DataPipelineConnection
method), 323
- describe_option_group_options()
(boto.rds.RDSConnection method), 640
- describe_option_group_options()
(boto.rds2.layer1.RDSConnection method),
894
- describe_option_groups() (boto.rds.RDSConnection
method), 640
- describe_option_groups()
(boto.rds2.layer1.RDSConnection method),
894
- describe_orderable_cluster_options()
(boto.redshift.layer1.RedshiftConnection
method), 670
- describe_orderable_db_instance_options()
(boto.rds2.layer1.RDSConnection method),
895

describe_permissions() (boto.opsworks.layer1.OpsWorksConnection method), 861

describe_pipelines() (boto.datapipeline.layer1.DataPipelineConnection method), 323

describe_raid_arrays() (boto.opsworks.layer1.OpsWorksConnection method), 862

describe_rank_expressions() (boto.cloudsearch.layer1.Layer1 method), 253

describe_rds_db_instances() (boto.opsworks.layer1.OpsWorksConnection method), 862

describe_replication_groups() (boto.elasticache.layer1.ElastiCacheConnection method), 479

describe_reserved_cache_nodes() (boto.elasticache.layer1.ElastiCacheConnection method), 479

describe_reserved_cache_nodes_offerings() (boto.elasticache.layer1.ElastiCacheConnection method), 480

describe_reserved_db_instances() (boto.rds2.layer1.RDSConnection method), 896

describe_reserved_db_instances_offerings() (boto.rds2.layer1.RDSConnection method), 896

describe_reserved_instances_modifications() (boto.ec2.connection.EC2Connection method), 414

describe_reserved_node_offerings() (boto.redshift.layer1.RedshiftConnection method), 671

describe_reserved_nodes() (boto.redshift.layer1.RedshiftConnection method), 671

describe_resize() (boto.redshift.layer1.RedshiftConnection method), 672

describe_resource() (boto.cloudformation.stack.Stack method), 233

describe_resources() (boto.cloudformation.stack.Stack method), 233

describe_scaling_parameters() (boto.cloudsearch2.layer1.CloudSearchConnection method), 270

describe_service_access_policies() (boto.cloudsearch.layer1.Layer1 method), 254

describe_service_access_policies() (boto.cloudsearch2.layer1.CloudSearchConnection method), 270

describe_service_errors() (boto.opsworks.layer1.OpsWorksConnection method), 862

describe_services() (boto.support.layer1.SupportConnection method), 930

describe_severity_levels() (boto.support.layer1.SupportConnection method), 930

describe_stack_events() (boto.cloudformation.connection.CloudFormationConnection method), 227

describe_stack_provisioning_parameters() (boto.opsworks.layer1.OpsWorksConnection method), 862

describe_stack_resource() (boto.cloudformation.connection.CloudFormationConnection method), 227

describe_stack_resources() (boto.cloudformation.connection.CloudFormationConnection method), 228

describe_stack_summary() (boto.opsworks.layer1.OpsWorksConnection method), 863

describe_stacks() (boto.cloudformation.connection.CloudFormationConnection method), 228

describe_stacks() (boto.opsworks.layer1.OpsWorksConnection method), 863

describe_stemming_options() (boto.cloudsearch.layer1.Layer1 method), 254

describe_step() (boto.emr.connection.EmrConnection method), 508

describe_stopword_options() (boto.cloudsearch.layer1.Layer1 method), 254

describe_stream() (boto.kinesis.layer1.KinesisConnection method), 590

describe_suggesters() (boto.cloudsearch2.layer1.CloudSearchConnection method), 270

describe_synonym_options() (boto.cloudsearch.layer1.Layer1 method), 254

describe_table() (boto.dynamodb.layer1.Layer1 method), 329

describe_table() (boto.dynamodb.layer2.Layer2 method), 333

describe_table() (boto.dynamodb2.layer1.DynamoDBConnection method), 379

describe_task_definition() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 465

describe_tasks() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 465

describe_time_based_auto_scaling() (boto.opsworks.layer1.OpsWorksConnection method), 863

describe_trails() (boto.cloudtrail.layer1.CloudTrailConnection method), 285

- describe_trusted_advisor_check_refresh_statuses() (boto.support.layer1.SupportConnection method), 930
- describe_trusted_advisor_check_result() (boto.support.layer1.SupportConnection method), 930
- describe_trusted_advisor_check_summaries() (boto.support.layer1.SupportConnection method), 931
- describe_trusted_advisor_checks() (boto.support.layer1.SupportConnection method), 931
- describe_user_profiles() (boto.opsworks.layer1.OpsWorksConnection method), 863
- describe_vault() (boto.glacier.layer1.Layer1 method), 524
- describe_volumes() (boto.opsworks.layer1.OpsWorksConnection method), 863
- describe_vpc_attribute() (boto.ec2.connection.EC2Connection method), 415
- describe_workflow_execution() (boto.swf.layer1.Layer1 method), 779
- describe_workflow_type() (boto.swf.layer1.Layer1 method), 780
- DescribeApplicationsResponse (class in boto.beanstalk.response), 203
- DescribeApplicationVersionsResponse (class in boto.beanstalk.response), 203
- DescribeAttachmentLimitExceeded, 932
- DescribeConfigurationOptionsResponse (class in boto.beanstalk.response), 203
- DescribeConfigurationSettingsResponse (class in boto.beanstalk.response), 203
- DescribeEnvironmentResourcesResponse (class in boto.beanstalk.response), 203
- DescribeEnvironmentsResponse (class in boto.beanstalk.response), 203
- DescribeEventsResponse (class in boto.beanstalk.response), 203
- DescriptionTooLongException, 304
- Destination (boto.mws.response.OrderReferenceDetails attribute), 628
- Destination (boto.mws.response.Subscription attribute), 630
- Destination (class in boto.mws.response), 623
- DestinationAddress (boto.mws.response.FulfillmentOrder attribute), 624
- DestinationList (boto.mws.response.ListRegisteredDestinationsResult attribute), 627
- detach() (boto.ec2.networkinterface.NetworkInterface method), 453
- detach() (boto.ec2.volume.Volume method), 461
- detach_classic_instance() (boto.vpc.vpc.VPC method), 819
- detach_classic_link_vpc() (boto.vpc.VPCConnection method), 809
- detach_elastic_load_balancer() (boto.opsworks.layer1.OpsWorksConnection method), 864
- detach_group_policy() (boto.iam.connection.IAMConnection method), 578
- detach_instances() (boto.ec2.autoscale.AutoScaleConnection method), 178
- detach_internet_gateway() (boto.vpc.VPCConnection method), 809
- detach_lb_from_subnets() (boto.ec2.elb.ELBConnection method), 499
- detach_network_interface() (boto.ec2.connection.EC2Connection method), 415
- detach_role_policy() (boto.iam.connection.IAMConnection method), 578
- detach_subnets() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
- detach_user_policy() (boto.iam.connection.IAMConnection method), 578
- detach_volume() (boto.ec2.connection.EC2Connection method), 415
- detach_vpn_gateway() (boto.vpc.VPCConnection method), 809
- Details (class in boto.ec2.instancestatus), 450
- DeveloperUserAlreadyRegisteredException, 311
- DhcpConfigSet (class in boto.vpc.dhcpoptions), 817
- DhcpOptions (class in boto.vpc.dhcpoptions), 817
- DhcpValueSet (class in boto.vpc.dhcpoptions), 817
- dict (boto.dynamodb.schema.Schema attribute), 344
- Dimension (class in boto.mws.response), 623
- disable() (boto.cloudfront.distribution.Distribution method), 236
- disable_actions() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- disable_alarm_actions() (boto.ec2.cloudwatch.CloudWatchConnection method), 290
- disable_availability_zones() (boto.ec2.elb.ELBConnection method), 499
- disable_classic_link() (boto.vpc.vpc.VPC method), 819
- disable_cross_zone_load_balancing() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
- disable_domain_transfer_lock() (boto.route53.domains.layer1.Route53DomainsConnection method), 911
- disable_hit() (boto.mturk.connection.MTurkConnection method), 602
- disable_key() (boto.kms.layer1.KMSConnection method), 824
- disable_key_rotation() (boto.kms.layer1.KMSConnection method), 824

- disable_logging() (boto.gs.bucket.Bucket method), 544
- disable_logging() (boto.redshift.layer1.RedshiftConnection method), 672
- disable_logging() (boto.s3.bucket.Bucket method), 701
- disable_metrics_collection() (boto.ec2.autoscale.AutoScaleConnection method), 178
- disable_snapshot_copy() (boto.redshift.layer1.RedshiftConnection method), 672
- disable_vgw_route_propagation() (boto.vpc.VPCCreation method), 810
- disable_vpc_classic_link() (boto.vpc.VPCCreation method), 810
- disable_zones() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
- DisabledException, 828
- disallow_doc_ip() (boto.cloudsearch.optionstatus.ServicePolicyStatus method), 258
- disallow_doc_ip() (boto.cloudsearch2.optionstatus.ServicePolicyStatus method), 273
- disallow_search_ip() (boto.cloudsearch.optionstatus.ServicePolicyStatus method), 258
- disallow_search_ip() (boto.cloudsearch2.optionstatus.ServicePolicyStatus method), 273
- disassociate() (boto.ec2.address.Address method), 404
- disassociate_address() (boto.ec2.connection.EC2Connection method), 415
- disassociate_elastic_ip() (boto.opsworks.layer1.OpsWorksConnection method), 864
- disassociate_network_acl() (boto.vpc.VPCCreation method), 810
- disassociate_route_table() (boto.vpc.VPCCreation method), 810
- discover_poll_endpoint() (boto.ec2.containerservice.layer1.EC2ContainerServiceConnection method), 465
- dispose_hit() (boto.mturk.connection.MTurkConnection method), 602
- dispose_qualification_type() (boto.mturk.connection.MTurkConnection method), 602
- Distribution (class in boto.cloudfront.distribution), 235
- DistributionConfig (class in boto.cloudfront.distribution), 237
- DistributionSummary (class in boto.cloudfront.distribution), 238
- DNSServerError, 691
- do_batches() (boto.services.bs.BS method), 743
- do_bool() (in module boto.cloudsearch.layer1), 256
- do_reset() (boto.services.bs.BS method), 743
- do_retrieve() (boto.services.bs.BS method), 743
- do_start() (boto.services.bs.BS method), 743
- do_status() (boto.services.bs.BS method), 743
- do_submit() (boto.services.bs.BS method), 743
- doc_service_arn (boto.cloudsearch.domain.Domain attribute), 249
- doc_service_endpoint (boto.cloudsearch.domain.Domain attribute), 249
- doc_service_endpoint (boto.cloudsearch2.domain.Domain attribute), 265
- DocumentServiceConnection (class in boto.cloudsearch.document), 261
- DocumentServiceConnection (class in boto.cloudsearch2.document), 276
- DocumentServiceException, 283
- domain (boto.sdb.db.manager.sdbmanager.SDBManager attribute), 921
- Domain (class in boto.cloudsearch.domain), 247
- Domain (class in boto.cloudsearch2.domain), 262
- Domain (class in boto.sdb.domain), 738
- Domain (class in boto.swf.layer2), 794
- domain_metadata() (boto.sdb.connection.SDBConnection method), 735
- DomainDumpParser (class in boto.sdb.domain), 740
- DomainMetadataExceeded, 918
- DomainMetaData (class in boto.sdb.domain), 741
- DownloadError (class in boto.glacier.concurrent.ConcurrentDownloader method), 538
- download_db_log_file_portion() (boto.rds2.layer1.RDSCreation method), 897
- download_to_file() (boto.glacier.job.Job method), 536
- download_to_fileobj() (boto.glacier.job.Job method), 537
- DownloadArchiveError, 539
- DownloadWorkerThread (class in boto.glacier.concurrent), 539
- dump() (boto.pyami.config.Config method), 630
- dump() (boto.sqs.queue.Queue method), 764
- dump_to_fileobj() (boto.pyami.config.Config method), 630
- dump_to_sdb() (boto.pyami.config.Config method), 630
- DuplicateRequest, 918
- duration_as_seconds() (boto.mturk.connection.MTurkConnection static method), 602
- dynamize_attribute_updates() (boto.dynamodb.layer2.Layer2 method), 333
- dynamize_expected_value() (boto.dynamodb.layer2.Layer2 method), 333
- dynamize_item() (boto.dynamodb.layer2.Layer2 method), 333
- dynamize_last_evaluated_key() (boto.dynamodb.layer2.Layer2 method), 333
- dynamize_range_key_condition() (boto.dynamodb.layer2.Layer2 method), 333
- dynamize_scan_filter() (boto.dynamodb.layer2.Layer2 method), 333

- method), 333
- dynamize_value() (in module boto.dynamodb.types), 348
- Dynamizer (class in boto.dynamodb.types), 347
- DynamoDBConnection (class in boto.dynamodb2.layer1), 369
- DynamoDBError, 402
- DynamoDBResponseError, 218
- E**
- Ebs (class in boto.ec2.autoscale.launchconfig), 184
- EBSBlockDeviceType (in module boto.ec2.blockdevicemapping), 404
- ec2_build_list_params() (boto.ec2.blockdevicemapping.BlockDeviceMapping method), 404
- EC2Connection (class in boto.ec2.connection), 405
- EC2ContainerServiceConnection (class in boto.ec2containerservice.layer1), 464
- Ec2InstanceAttributes (class in boto.emr.emrobject), 513
- EC2Object (class in boto.ec2.ec2object), 440
- EC2RegionInfo (class in boto.ec2.regioninfo), 454
- EC2ResponseError, 218
- EC2SecurityGroup (class in boto.rds.dbsecuritygroup), 652
- ECSCConnection (class in boto.ecs), 470
- ElastiCacheConnection (class in boto.elasticache.layer1), 471
- ElasticTranscoderConnection (class in boto.elastictranscoder.layer1), 485
- ELBConnection (class in boto.ec2.elb), 495
- Element (class in boto.mws.response), 623
- element_factory() (boto.mws.connection.MWSConnection.ResponseFactory method), 611
- element_factory() (boto.mws.response.ResponseFactory method), 629
- ElementList (class in boto.mws.response), 624
- emit() (boto.NullHandler method), 204
- emit() (boto.utils.AuthSMTPHandler method), 221
- empty() (boto.sdb.db.property.BooleanProperty method), 921
- empty() (boto.sdb.db.property.FloatProperty method), 922
- empty() (boto.sdb.db.property.IntegerProperty method), 923
- empty() (boto.sdb.db.property.ListProperty method), 923
- empty() (boto.sdb.db.property.LongProperty method), 923
- empty() (boto.sdb.db.property.MapProperty method), 923
- empty() (boto.sdb.db.property.Property method), 924
- EmrConnection (class in boto.emr.connection), 506
- EmrObject (class in boto.emr.emrobject), 513
- EmrResponseError, 218
- enable() (boto.cloudfront.distribution.Distribution method), 236
- enable_actions() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
- enable_alarm_actions() (boto.ec2.cloudwatch.CloudWatchConnection method), 290
- enable_availability_zones() (boto.ec2.elb.ELBConnection method), 499
- enable_classic_link() (boto.vpc.vpc.VPC method), 819
- enable_cross_zone_load_balancing() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
- enable_domain_transfer_lock() (boto.route53.domains.layer1.Route53DomainsConnection method), 911
- enable_key() (boto.kms.layer1.KMSConnection method), 824
- enable_key_rotation() (boto.kms.layer1.KMSConnection method), 824
- enable_logging() (boto.gs.bucket.Bucket method), 544
- enable_logging() (boto.redshift.layer1.RedshiftConnection method), 672
- enable_logging() (boto.s3.bucket.Bucket method), 701
- enable_metrics_collection() (boto.ec2.autoscale.AutoScaleConnection method), 178
- enable_mfa_device() (boto.iam.connection.IAMConnection method), 578
- enable_snapshot_copy() (boto.redshift.layer1.RedshiftConnection method), 673
- enable_vgw_route_propagation() (boto.vpc.VPCConnection method), 810
- enable_volume_io() (boto.ec2.connection.EC2Connection method), 416
- enable_vpc_classic_link() (boto.vpc.VPCConnection method), 811
- enable_zones() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
- EnabledMetric (class in boto.ec2.autoscale.group), 183
- encode() (boto.dynamodb.types.Binary method), 347
- encode() (boto.dynamodb.types.Dynamizer method), 348
- encode() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
- encode() (boto.sqs.jsonmessage.JSONMessage method), 761
- encode() (boto.sqs.message.EncodedMHMessage method), 762
- encode() (boto.sqs.message.Message method), 763
- encode() (boto.sqs.message.MHMessage method), 762
- encode() (boto.sqs.message.RawMessage method), 763
- encode_blob() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920
- encode_bool() (boto.cloudformation.connection.CloudFormationConnection method), 228
- encode_bool() (boto.sdb.db.manager.sdbmanager.SDBConverter method), 920

encode_date() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_datetime() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_float() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_int() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_list() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_long() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_map() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_prop() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_reference() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_string() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_time() (boto.sdb.db.manager.sdbmanager.SDBConverterElement() method), 920
 encode_value() (boto.sdb.db.manager.sdbmanager.SDBManagerElement() method), 921
 EncodedMHMessage (class in boto.sqs.message), 762
 EncodingError, 262, 277
 encrypt() (boto.kms.layer1.KMSConnection method), 824
 EncryptionConfigBody (boto.gs.bucket.Bucket attribute), 541
 EncryptionConfigDefaultKeyNameFragment (boto.gs.bucket.Bucket attribute), 541
 end() (boto.mws.response.DeclarativeType method), 623
 end() (boto.mws.response.Element method), 623
 end() (boto.mws.response.ElementList method), 624
 end() (boto.mws.response.SimpleList method), 629
 endElement() (boto.cloudformation.stack.Capability method), 232
 endElement() (boto.cloudformation.stack.NotificationARN method), 232
 endElement() (boto.cloudformation.stack.Output method), 232
 endElement() (boto.cloudformation.stack.Parameter method), 232
 endElement() (boto.cloudformation.stack.Stack method), 233
 endElement() (boto.cloudformation.stack.StackEvent method), 233
 endElement() (boto.cloudformation.stack.StackResource method), 233
 endElement() (boto.cloudformation.stack.StackResourceSummary method), 233
 endElement() (boto.cloudformation.stack.StackSummary method), 234
 endElement() (boto.cloudformation.stack.Tag method), 234
 endElement() (boto.cloudformation.template.Template method), 234
 endElement() (boto.cloudformation.template.TemplateParameter method), 234
 endElement() (boto.cloudfront.distribution.Distribution method), 236
 endElement() (boto.cloudfront.distribution.DistributionConfig method), 238
 endElement() (boto.cloudfront.distribution.DistributionSummary method), 238
 endElement() (boto.cloudfront.identity.OriginAccessIdentity method), 240
 endElement() (boto.cloudfront.identity.OriginAccessIdentityConfig method), 240
 endElement() (boto.cloudfront.identity.OriginAccessIdentitySummary method), 240
 endElement() (boto.cloudfront.invalidation.InvalidationBatch method), 241
 endElement() (boto.cloudfront.invalidation.InvalidationListResultSet method), 242
 endElement() (boto.cloudfront.invalidation.InvalidationSummary method), 242
 endElement() (boto.cloudfront.logging.LoggingInfo method), 243
 endElement() (boto.cloudfront.origin.CustomOrigin method), 240
 endElement() (boto.cloudfront.origin.S3Origin method), 240
 endElement() (boto.cloudfront.signers.ActiveTrustedSigners method), 241
 endElement() (boto.cloudfront.signers.Signer method), 241
 endElement() (boto.cloudfront.signers.TrustedSigners method), 241
 endElement() (boto.cloudsearch.optionstatus.OptionStatus method), 257
 endElement() (boto.ec2.address.Address method), 404
 endElement() (boto.ec2.autoscale.activity.Activity method), 182
 endElement() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 endElement() (boto.ec2.autoscale.group.AutoScalingGroupMetric method), 183
 endElement() (boto.ec2.autoscale.group.EnabledMetric method), 183
 endElement() (boto.ec2.autoscale.group.ProcessType method), 183
 endElement() (boto.ec2.autoscale.group.SuspendedProcess method), 184
 endElement() (boto.ec2.autoscale.group.TerminationPolicies method), 184
 endElement() (boto.ec2.autoscale.instance.Instance

method), 184
endElement() (boto.ec2.autoscale.launchconfig.BlockDeviceMapping method), 502
method), 184
endElement() (boto.ec2.autoscale.launchconfig.Ebs method), 184
endElement() (boto.ec2.autoscale.launchconfig.InstanceMonitoring method), 184
endElement() (boto.ec2.autoscale.launchconfig.LaunchConfiguration method), 186
endElement() (boto.ec2.autoscale.policy.AdjustmentType method), 186
endElement() (boto.ec2.autoscale.policy.Alarm method), 186
endElement() (boto.ec2.autoscale.policy.MetricCollectionTypes method), 186
endElement() (boto.ec2.autoscale.policy.MetricCollectionTypesBaseType method), 186
endElement() (boto.ec2.autoscale.policy.ScalingPolicy method), 187
endElement() (boto.ec2.autoscale.policy.TerminationPolicies method), 187
endElement() (boto.ec2.autoscale.request.Request method), 187
endElement() (boto.ec2.autoscale.scheduled.ScheduledUpdateGroupAction method), 187
endElement() (boto.ec2.autoscale.tag.Tag method), 188
endElement() (boto.ec2.blockdevicemapping.BlockDeviceMapping method), 404
endElement() (boto.ec2.blockdevicemapping.BlockDeviceType method), 404
endElement() (boto.ec2.cloudwatch.alarm.AlarmHistoryItem method), 294
endElement() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
endElement() (boto.ec2.cloudwatch.alarm.MetricAlarms method), 295
endElement() (boto.ec2.cloudwatch.datapoint.Datapoint method), 292
endElement() (boto.ec2.cloudwatch.metric.Metric method), 293
endElement() (boto.ec2.ec2object.EC2Object method), 440
endElement() (boto.ec2.elb.attributes.AccessLogAttribute method), 506
endElement() (boto.ec2.elb.attributes.ConnectionDrainingAttribute method), 506
endElement() (boto.ec2.elb.attributes.ConnectionSettingAttribute method), 506
endElement() (boto.ec2.elb.attributes.CrossZoneLoadBalancingAttribute method), 506
endElement() (boto.ec2.elb.attributes.LbAttributes method), 506
endElement() (boto.ec2.elb.healthcheck.HealthCheck method), 501
endElement() (boto.ec2.elb.instancestate.InstanceState method), 502
endElement() (boto.ec2.elb.listelement.ListElement method), 502
endElement() (boto.ec2.elb.listener.Listener method), 502
endElement() (boto.ec2.elb.loadbalancer.Backend method), 502
endElement() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504
endElement() (boto.ec2.elb.loadbalancer.LoadBalancerZones method), 505
endElement() (boto.ec2.elb.policies.AppCookieStickinessPolicy method), 505
endElement() (boto.ec2.elb.policies.LBCookieStickinessPolicy method), 505
endElement() (boto.ec2.elb.policies.OtherPolicy method), 505
endElement() (boto.ec2.elb.policies.Policies method), 505
endElement() (boto.ec2.elb.securitygroup.SecurityGroup method), 505
endElement() (boto.ec2.group.Group method), 441
endElement() (boto.ec2.image.BillingProducts method), 441
endElement() (boto.ec2.image.CopyImage method), 441
endElement() (boto.ec2.image.Image method), 441
endElement() (boto.ec2.image.ImageAttribute method), 444
endElement() (boto.ec2.image.ProductCodes method), 444
endElement() (boto.ec2.instance.ConsoleOutput method), 444
endElement() (boto.ec2.instance.Instance method), 446
endElement() (boto.ec2.instance.InstanceAttribute method), 448
endElement() (boto.ec2.instance.InstancePlacement method), 448
endElement() (boto.ec2.instance.InstanceState method), 449
endElement() (boto.ec2.instance.Reservation method), 449
endElement() (boto.ec2.instance.SubParse method), 449
endElement() (boto.ec2.instanceinfo.InstanceInfo method), 449
endElement() (boto.ec2.instancestatus.Details method), 450
endElement() (boto.ec2.instancestatus.Event method), 450
endElement() (boto.ec2.instancestatus.EventSet method), 450
endElement() (boto.ec2.instancestatus.InstanceStatus method), 450
endElement() (boto.ec2.instancestatus.InstanceStatusSet method), 450

endElement() (boto.ec2.instancestatus.Status method), 451
 endElement() (boto.ec2.keypair.KeyPair method), 451
 endElement() (boto.ec2.launchspecification.GroupList method), 451
 endElement() (boto.ec2.launchspecification.LaunchSpecification method), 452
 endElement() (boto.ec2.networkinterface.Attachment method), 452
 endElement() (boto.ec2.networkinterface.NetworkInterface method), 453
 endElement() (boto.ec2.networkinterface.PrivateIPAddress method), 453
 endElement() (boto.ec2.placementgroup.PlacementGroup method), 454
 endElement() (boto.ec2.reservedinstance.InstanceCount method), 454
 endElement() (boto.ec2.reservedinstance.ModificationResult method), 454
 endElement() (boto.ec2.reservedinstance.ModifyReservedInstancesResult method), 454
 endElement() (boto.ec2.reservedinstance.PriceSchedule method), 454
 endElement() (boto.ec2.reservedinstance.PricingDetail method), 454
 endElement() (boto.ec2.reservedinstance.RecurringCharge method), 455
 endElement() (boto.ec2.reservedinstance.ReservedInstance method), 455
 endElement() (boto.ec2.reservedinstance.ReservedInstanceListing method), 455
 endElement() (boto.ec2.reservedinstance.ReservedInstancesConfiguration method), 455
 endElement() (boto.ec2.reservedinstance.ReservedInstancesModification method), 455
 endElement() (boto.ec2.reservedinstance.ReservedInstancesOffering method), 456
 endElement() (boto.ec2.securitygroup.GroupOrCIDR method), 456
 endElement() (boto.ec2.securitygroup.IPPermissions method), 456
 endElement() (boto.ec2.securitygroup.IPPermissionsList method), 456
 endElement() (boto.ec2.securitygroup.SecurityGroup method), 457
 endElement() (boto.ec2.snapshot.Snapshot method), 458
 endElement() (boto.ec2.snapshot.SnapshotAttribute method), 458
 endElement() (boto.ec2.spotinstancerequest.SpotInstanceRequest method), 459
 endElement() (boto.ec2.spotinstancerequest.SpotInstanceStateFault method), 459
 endElement() (boto.ec2.spotinstancerequest.SpotInstanceStatus method), 459
 endElement() (boto.ec2.tag.Tag method), 460
 endElement() (boto.ec2.tag.TagSet method), 460
 endElement() (boto.ec2.volume.AttachmentSet method), 460
 endElement() (boto.ec2.volume.Volume method), 461
 endElement() (boto.ec2.volume.VolumeAttribute method), 462
 endElement() (boto.ec2.volumestatus.Action method), 462
 endElement() (boto.ec2.volumestatus.ActionSet method), 462
 endElement() (boto.ec2.volumestatus.Event method), 462
 endElement() (boto.ec2.volumestatus.EventSet method), 463
 endElement() (boto.ec2.volumestatus.VolumeStatus method), 463
 endElement() (boto.ec2.volumestatus.VolumeStatusSet method), 463
 endElement() (boto.ec2.zone.MessageSet method), 463
 endElement() (boto.ec2.zone.Zone method), 463
 endElement() (boto.ecs.item.ItemSet method), 470
 endElement() (boto.ecs.item.ResponseGroup method), 470
 endElement() (boto.emr.emrobject.Arg method), 513
 endElement() (boto.emr.emrobject.EmrObject method), 514
 endElement() (boto.exception.BotoServerError method), 218
 endElement() (boto.exception.ConsoleOutput method), 218
 endElement() (boto.exception.EC2ResponseError method), 218
 endElement() (boto.exception.SQSErrorMessage method), 220
 endElement() (boto.exception.StorageCreateError method), 220
 endElement() (boto.exception.StorageResponseError method), 220
 endElement() (boto.gs.acl.ACL method), 540
 endElement() (boto.gs.acl.Entries method), 540
 endElement() (boto.gs.acl.Entry method), 540
 endElement() (boto.gs.acl.Scope method), 540
 endElement() (boto.gs.cors.Cors method), 559
 endElement() (boto.gs.key.Key method), 562
 endElement() (boto.gs.user.User method), 572
 endElement() (boto.handler.XmlHandler method), 220
 endElement() (boto.iam.summarymap.SummaryMap method), 588
 endElement() (boto.mturk.connection.Assignment method), 600
 endElement() (boto.mturk.connection.BaseAutoResultElement method), 600
 endElement() (boto.mturk.connection.QualificationRequest method), 605
 endElement() (boto.mturk.connection.QuestionFormAnswer

- method), 605
- endElement() (boto.mturk.price.Price method), 606
- endElement() (boto.mws.response.ComplexAmount method), 623
- endElement() (boto.mws.response.ComplexDimensions method), 623
- endElement() (boto.mws.response.ComplexWeight method), 623
- endElement() (boto.mws.response.ResponseElement method), 629
- endElement() (boto.rds.dbinstance.DBInstance method), 649
- endElement() (boto.rds.dbinstance.PendingModifiedValues method), 651
- endElement() (boto.rds.dbinstance.ReadReplicaDBInstance method), 651
- endElement() (boto.rds.dbsecuritygroup.DBSecurityGroup method), 652
- endElement() (boto.rds.dbsecuritygroup.EC2SecurityGroup method), 652
- endElement() (boto.rds.dbsecuritygroup.IPRange method), 652
- endElement() (boto.rds.dbsnapshot.DBSnapshot method), 653
- endElement() (boto.rds.event.Event method), 653
- endElement() (boto.rds.parametergroup.Parameter method), 653
- endElement() (boto.rds.parametergroup.ParameterGroup method), 654
- endElement() (boto.resultset.BooleanResult method), 220
- endElement() (boto.resultset.ResultSet method), 221
- endElement() (boto.route53.hostedzone.HostedZone method), 693
- endElement() (boto.route53.record.Record method), 693
- endElement() (boto.route53.record.ResourceRecordSets method), 695
- endElement() (boto.s3.acl.ACL method), 697
- endElement() (boto.s3.acl.Grant method), 698
- endElement() (boto.s3.acl.Policy method), 698
- endElement() (boto.s3.bucket.Bucket method), 701
- endElement() (boto.s3.cors.CORSConfiguration method), 715
- endElement() (boto.s3.cors.CORSRule method), 715
- endElement() (boto.s3.deletemarker.DeleteMarker method), 716
- endElement() (boto.s3.key.Key method), 718
- endElement() (boto.s3.lifecycle.Expiration method), 730
- endElement() (boto.s3.lifecycle.Lifecycle method), 731
- endElement() (boto.s3.lifecycle.Rule method), 731
- endElement() (boto.s3.lifecycle.Transitions method), 732
- endElement() (boto.s3.multidelete.Deleted method), 728
- endElement() (boto.s3.multidelete.Error method), 729
- endElement() (boto.s3.multidelete.MultiDeleteResult method), 729
- endElement() (boto.s3.multipart.CompleteMultiPartUpload method), 727
- endElement() (boto.s3.multipart.MultiPartUpload method), 727
- endElement() (boto.s3.multipart.Part method), 728
- endElement() (boto.s3.prefix.Prefix method), 727
- endElement() (boto.s3.tagging.Tag method), 732
- endElement() (boto.s3.tagging.Tags method), 732
- endElement() (boto.s3.tagging.TagSet method), 732
- endElement() (boto.s3.user.User method), 732
- endElement() (boto.sdb.domain.Domain method), 739
- endElement() (boto.sdb.domain.DomainDumpParser method), 741
- endElement() (boto.sdb.domain.DomainMetaDataIdentifiers method), 741
- endElement() (boto.sdb.item.Item method), 742
- endElement() (boto.sqs.attributes.Attributes method), 755
- endElement() (boto.sqs.batchresults.BatchResults method), 769
- endElement() (boto.sqs.batchresults.ResultEntry method), 769
- endElement() (boto.sqs.message.RawMessage method), 763
- endElement() (boto.sqs.queue.Queue method), 764
- endElement() (boto.sts.credentials.AssumedRole method), 775
- endElement() (boto.sts.credentials.Credentials method), 775
- endElement() (boto.sts.credentials.DecodeAuthorizationMessage method), 776
- endElement() (boto.sts.credentials.FederationToken method), 776
- endElement() (boto.sts.credentials.User method), 776
- endElement() (boto.vpc.customergateway.CustomerGateway method), 817
- endElement() (boto.vpc.dhcpoptions.DhcpConfigSet method), 817
- endElement() (boto.vpc.dhcpoptions.DhcpOptions method), 817
- endElement() (boto.vpc.dhcpoptions.DhcpValueSet method), 817
- endElement() (boto.vpc.internetgateway.InternetGateway method), 817
- endElement() (boto.vpc.internetgateway.InternetGatewayAttachment method), 818
- endElement() (boto.vpc.routetable.Route method), 818
- endElement() (boto.vpc.routetable.RouteAssociation method), 818
- endElement() (boto.vpc.routetable.RouteTable method), 818
- endElement() (boto.vpc.subnet.Subnet method), 818
- endElement() (boto.vpc.vpc.VPC method), 820
- endElement() (boto.vpc.vpnconnection.VpnConnection method), 820

- endElement() (boto.vpc.vpnconnection.VpnConnectionOptions method), 820
 - endElement() (boto.vpc.vpnconnection.VpnStaticRoute method), 821
 - endElement() (boto.vpc.vpnconnection.VpnTunnel method), 821
 - endElement() (boto.vpc.vpngateway.Attachment method), 821
 - endElement() (boto.vpc.vpngateway.VpnGateway method), 821
 - endNode() (boto.sqs.message.RawMessage method), 763
 - Entries (class in boto.gs.acl), 540
 - Entry (class in boto.gs.acl), 540
 - EnvironmentDescription (class in boto.beanstalk.response), 203
 - EnvironmentInfoDescription (class in boto.beanstalk.response), 203
 - EnvironmentResourceDescription (class in boto.beanstalk.response), 203
 - EnvironmentResourcesDescription (class in boto.beanstalk.response), 203
 - Error (boto.mws.response.ProductsBulkOperationResult attribute), 628
 - Error (class in boto.s3.multidelete), 728
 - errors (boto.cloudsearch2.document.CommitMismatchError attribute), 276
 - escape() (boto.cloudfront.invalidation.InvalidationBatch method), 241
 - estimate_template_cost() (boto.cloudformation.connection.CloudFormationConnection method), 228
 - EstimatedFees (boto.mws.response.FulfillmentPreview attribute), 624
 - EstimatedShippingWeight (boto.mws.response.FulfillmentPreview attribute), 624
 - EstimatedShippingWeight (boto.mws.response.FulfillmentPreviewItem attribute), 624
 - EU (boto.gs.connection.Location attribute), 559
 - EU (boto.s3.connection.Location attribute), 711
 - EUCentral1 (boto.s3.connection.Location attribute), 711
 - evaluate_expression() (boto.datapipeline.layer1.DataPipelineConnection method), 323
 - EvaluateTargetHealth (boto.route53.record.Record attribute), 693
 - Event (class in boto.ec2.instancestatus), 450
 - Event (class in boto.ec2.volumestatus), 462
 - Event (class in boto.mturk.notification), 605
 - Event (class in boto.rds.event), 653
 - EVENT_PATTERN (boto.mturk.notification.NotificationMessage attribute), 605
 - EVENT_RE (boto.mturk.notification.NotificationMessage attribute), 605
 - EventDescription (class in boto.beanstalk.response), 203
 - EventSet (class in boto.ec2.instancestatus), 450
 - EventSet (class in boto.ec2.volumestatus), 463
 - EventSubscriptionQuotaExceeded, 684, 875
 - execute_policy() (boto.ec2.autoscale.AutoScaleConnection method), 178
 - executions() (boto.swf.layer2.Domain method), 796
 - exists() (boto.gs.key.Key method), 562
 - exists() (boto.manage.cmdshell.LocalClient method), 597
 - exists() (boto.manage.cmdshell.SSHClient method), 598
 - exists() (boto.s3.key.Key method), 718
 - Expiration (class in boto.s3.lifecycle), 730
 - expire_hit() (boto.mturk.connection.MTurkConnection method), 602
 - expired (boto.mturk.connection.HIT attribute), 601
 - ExpiredIteratorException, 596
 - ExpressionStatus (class in boto.cloudsearch2.optionstatus), 272
 - extend_hit() (boto.mturk.connection.MTurkConnection method), 602
 - ExternalQuestion (class in boto.mturk.question), 608
- ## F
- f (boto.gs.key.Key attribute), 562
 - f (boto.s3.key.Key attribute), 718
 - fail() (boto.swf.layer2.ActivityWorker method), 793
 - fail_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 790
 - FollowerBody (boto.route53.record.Record attribute), 693
 - FakeServer (class in boto.manage.cmdshell), 597
 - FederationToken (class in boto.sts.credentials), 776
 - FeedSubmissionInfo (boto.mws.response.GetFeedSubmissionListResult attribute), 624
 - FeedSubmissionInfo (boto.mws.response.SubmitFeedResult attribute), 630
 - FeedSubmissionInfo (class in boto.mws.response), 624
 - FeeRefunded (boto.mws.response.RefundDetails attribute), 629
 - fetch() (boto.sdb.db.query.Query method), 925
 - fetch_file() (in module boto.utils), 223
 - fetch_more() (boto.dynamodb2.results.BatchGetResultSet method), 356
 - fetch_more() (boto.dynamodb2.results.ResultSet method), 356
 - fetch_s3_file() (boto.pyami.bootstrap.Bootstrap method), 630
 - Fields (boto.emr.emrobject.AddInstanceGroupsResponse attribute), 512
 - Fields (boto.emr.emrobject.Application attribute), 513
 - Fields (boto.emr.emrobject.BootstrapAction attribute), 513
 - Fields (boto.emr.emrobject.BootstrapActionList attribute), 513

- Fields (boto.emr.emrobject.Cluster attribute), 513
 - Fields (boto.emr.emrobject.ClusterStateChangeReason attribute), 513
 - Fields (boto.emr.emrobject.ClusterStatus attribute), 513
 - Fields (boto.emr.emrobject.ClusterSummary attribute), 513
 - Fields (boto.emr.emrobject.ClusterSummaryList attribute), 513
 - Fields (boto.emr.emrobject.ClusterTimeline attribute), 513
 - Fields (boto.emr.emrobject.Ec2InstanceAttributes attribute), 513
 - Fields (boto.emr.emrobject.EmrObject attribute), 514
 - Fields (boto.emr.emrobject.HadoopStep attribute), 514
 - Fields (boto.emr.emrobject.InstanceGroup attribute), 514
 - Fields (boto.emr.emrobject.InstanceGroupInfo attribute), 514
 - Fields (boto.emr.emrobject.InstanceGroupList attribute), 514
 - Fields (boto.emr.emrobject.InstanceInfo attribute), 514
 - Fields (boto.emr.emrobject.InstanceList attribute), 514
 - Fields (boto.emr.emrobject.JobFlow attribute), 514
 - Fields (boto.emr.emrobject.KeyValue attribute), 514
 - Fields (boto.emr.emrobject.ModifyInstanceGroupsResponse attribute), 514
 - Fields (boto.emr.emrobject.RunJobFlowResponse attribute), 515
 - Fields (boto.emr.emrobject.Step attribute), 515
 - Fields (boto.emr.emrobject.StepConfig attribute), 515
 - Fields (boto.emr.emrobject.StepSummary attribute), 515
 - Fields (boto.emr.emrobject.StepSummaryList attribute), 515
 - file (boto.sdb.db.blob.Blob attribute), 918
 - FileConnection (class in boto.file.connection), 516
 - FileUploadAnswer (class in boto.mturk.question), 608
 - FileUploadURL (class in boto.mturk.connection), 600
 - filter() (boto.sdb.db.query.Query method), 925
 - find_class() (in module boto.utils), 223
 - find_element() (boto.mws.connection.MWSConnection.ResponseFactory method), 611
 - find_element() (boto.mws.response.ResponseFactory method), 629
 - find_matching_headers() (in module boto.utils), 223
 - find_records() (boto.route53.zone.Zone method), 696
 - first_key (boto.dynamodb2.results.ResultSet attribute), 356
 - Flash (class in boto.mturk.question), 608
 - float_to_decimal() (in module boto.dynamodb.types), 348
 - FloatProperty (class in boto.sdb.db.property), 922
 - flush() (boto.dynamodb2.table.BatchTable method), 357
 - for_key() (boto.services.message.ServiceMessage method), 743
 - FormattedContent (class in boto.mturk.question), 608
 - FPSConnection (class in boto.fps.connection), 518
 - FreeTextAnswer (class in boto.mturk.question), 608
 - from_json() (boto.sts.credentials.Credentials class method), 775
 - from_path() (boto.sdb.db.key.Key class method), 919
 - from_xml() (boto.sdb.domain.Domain method), 739
 - FulfillmentOrder (boto.mws.response.GetFulfillmentOrderResult attribute), 624
 - FulfillmentOrder (class in boto.mws.response), 624
 - FulfillmentOrderItem (boto.mws.response.GetFulfillmentOrderResult attribute), 624
 - FulfillmentOrders (boto.mws.response.ListAllFulfillmentOrdersResult attribute), 626
 - FulfillmentPreview (class in boto.mws.response), 624
 - FulfillmentPreviewItem (class in boto.mws.response), 624
 - FulfillmentPreviews (boto.mws.response.GetFulfillmentPreviewResult attribute), 625
 - FulfillmentPreviewShipments (boto.mws.response.FulfillmentPreview attribute), 624
 - FulfillmentShipment (boto.mws.response.GetFulfillmentOrderResult attribute), 625
 - fund_prepaid() (boto.fps.connection.FPSConnection method), 518
- ## G
- GemType (boto.mws.response.VariationRelationship attribute), 630
 - generate_credential_report() (boto.iam.connection.IAMConnection method), 578
 - generate_data_key() (boto.kms.layer1.KMSConnection method), 825
 - generate_data_key_without_plaintext() (boto.kms.layer1.KMSConnection method), 825
 - generate_parts_from_fobj() (in module boto.glacier.writer), 537
 - generate_random() (boto.kms.layer1.KMSConnection method), 825
 - generate_url() (boto.gs.bucket.Bucket method), 544
 - generate_url() (boto.gs.connection.GSConnection method), 557
 - generate_url() (boto.gs.key.Key method), 562
 - generate_url() (boto.s3.bucket.Bucket method), 701
 - generate_url() (boto.s3.connection.S3Connection method), 712
 - generate_url() (boto.s3.key.Key method), 718
 - generate_url_sigv4() (boto.gs.connection.GSConnection method), 557
 - generate_url_sigv4() (boto.s3.connection.S3Connection method), 712
 - get() (boto.connection.HostConnectionPool method), 217
 - get() (boto.dynamodb2.items.Item method), 354

[get\(\)](#) (boto.ec2.buyreservation.BuyReservation method), 404
[get\(\)](#) (boto.ecs.item.ResponseGroup method), 470
[get\(\)](#) (boto.pyami.config.Config method), 630
[get\(\)](#) (boto.services.servicedef.ServiceDef method), 744
[get\(\)](#) (boto.sqs.message.MHMessage method), 762
[get\(\)](#) (boto.utils.LazyLoadMetadata method), 222
[get\(\)](#) (in module boto.manage.propget), 600
[get_a\(\)](#) (boto.route53.zone.Zone method), 696
[get_access_policies\(\)](#) (boto.cloudsearch.domain.Domain method), 249
[get_access_policies\(\)](#) (boto.cloudsearch2.domain.Domain method), 265
[get_account_activity\(\)](#) (boto.fps.connection.FPSConnection method), 518
[get_account_alias\(\)](#) (boto.iam.connection.IAMConnection method), 578
[get_account_balance\(\)](#) (boto.fps.connection.FPSConnection method), 518
[get_account_balance\(\)](#) (boto.mturk.connection.MTurkConnection method), 602
[get_account_limits\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 178
[get_account_password_policy\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_account_summary\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_acl\(\)](#) (boto.gs.bucket.Bucket method), 544
[get_acl\(\)](#) (boto.gs.key.Key method), 563
[get_acl\(\)](#) (boto.s3.bucket.Bucket method), 702
[get_acl\(\)](#) (boto.s3.key.Key method), 719
[get_activities\(\)](#) (boto.ec2.autoscale.group.AutoScalingGroup method), 183
[get_all_access_keys\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_all_activities\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 178
[get_all_addresses\(\)](#) (boto.ec2.connection.EC2Connection method), 416
[get_all_adjustment_types\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 178
[get_all_autoscaling_instances\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 178
[get_all_buckets\(\)](#) (boto.gs.connection.GSConnection method), 557
[get_all_buckets\(\)](#) (boto.s3.connection.S3Connection method), 712
[get_all_bundle_tasks\(\)](#) (boto.ec2.connection.EC2Connection method), 416
[get_all_classic_link_instances\(\)](#) (boto.ec2.connection.EC2Connection method), 416
[get_all_classic_link_vpcs\(\)](#) (boto.vpc.VPCConnection method), 811
[get_all_customer_gateways\(\)](#) (boto.vpc.VPCConnection method), 811
[get_all_db_subnet_groups\(\)](#) (boto.rds.RDSConnection method), 640
[get_all_dbinstances\(\)](#) (boto.rds.RDSConnection method), 641
[get_all_dbparameter_groups\(\)](#) (boto.rds.RDSConnection method), 641
[get_all_dbparameters\(\)](#) (boto.rds.RDSConnection method), 641
[get_all_dbsecurity_groups\(\)](#) (boto.rds.RDSConnection method), 642
[get_all_dbsnapshots\(\)](#) (boto.rds.RDSConnection method), 642
[get_all_dhcp_options\(\)](#) (boto.vpc.VPCConnection method), 811
[get_all_distribution\(\)](#) (boto.cloudfront.CloudFrontConnection method), 235
[get_all_domains\(\)](#) (boto.sdb.connection.SDBConnection method), 735
[get_all_events\(\)](#) (boto.rds.RDSConnection method), 642
[get_all_group_policies\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_all_groups\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 179
[get_all_groups\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_all_hits\(\)](#) (boto.cloudsearch.search.SearchConnection method), 258
[get_all_hits\(\)](#) (boto.cloudsearch2.search.SearchConnection method), 274
[get_all_hits\(\)](#) (boto.mturk.connection.MTurkConnection method), 602
[get_all_hosted_zones\(\)](#) (boto.route53.connection.Route53Connection method), 690
[get_all_images\(\)](#) (boto.ec2.connection.EC2Connection method), 417
[get_all_instance_status\(\)](#) (boto.ec2.connection.EC2Connection method), 417
[get_all_instance_types\(\)](#) (boto.ec2.connection.EC2Connection method), 417
[get_all_instances\(\)](#) (boto.ec2.connection.EC2Connection method), 418
[get_all_internet_gateways\(\)](#) (boto.vpc.VPCConnection method), 812
[get_all_kernels\(\)](#) (boto.ec2.connection.EC2Connection method), 418
[get_all_key_pairs\(\)](#) (boto.ec2.connection.EC2Connection method), 418
[get_all_keys\(\)](#) (boto.file.bucket.Bucket method), 515
[get_all_keys\(\)](#) (boto.gs.bucket.Bucket method), 544

- [get_all_keys\(\)](#) (boto.s3.bucket.Bucket method), 702
[get_all_launch_configurations\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 179
[get_all_lb_attributes\(\)](#) (boto.ec2.elb.ELBConnection method), 499
[get_all_load_balancers\(\)](#) (boto.ec2.elb.ELBConnection method), 499
[get_all_logs\(\)](#) (boto.rds.RDSConnection method), 643
[get_all_metric_collection_types\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 179
[get_all_mfa_devices\(\)](#) (boto.iam.connection.IAMConnection method), 579
[get_all_multipart_uploads\(\)](#) (boto.gs.bucket.Bucket method), 545
[get_all_multipart_uploads\(\)](#) (boto.s3.bucket.Bucket method), 702
[get_all_network_acls\(\)](#) (boto.vpc.VPCConnection method), 812
[get_all_network_interfaces\(\)](#) (boto.ec2.connection.EC2Connection method), 418
[get_all_origin_access_identity\(\)](#) (boto.cloudfront.CloudFrontConnection method), 235
[get_all_paged\(\)](#) (boto.cloudsearch.search.SearchConnection method), 259
[get_all_paged\(\)](#) (boto.cloudsearch2.search.SearchConnection method), 274
[get_all_parts\(\)](#) (boto.s3.multipart.MultiPartUpload method), 727
[get_all_placement_groups\(\)](#) (boto.ec2.connection.EC2Connection method), 419
[get_all_policies\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 179
[get_all_qualifications_for_qual_type\(\)](#) (boto.mturk.connection.MTurkConnection method), 602
[get_all_queues\(\)](#) (boto.sqs.connection.SQSConnection method), 757
[get_all_ramdisks\(\)](#) (boto.ec2.connection.EC2Connection method), 419
[get_all_regions\(\)](#) (boto.ec2.connection.EC2Connection method), 419
[get_all_reservations\(\)](#) (boto.ec2.connection.EC2Connection method), 419
[get_all_reserved_instances\(\)](#) (boto.ec2.connection.EC2Connection method), 420
[get_all_reserved_instances_offerings\(\)](#) (boto.ec2.connection.EC2Connection method), 420
[get_all_route_tables\(\)](#) (boto.vpc.VPCConnection method), 812
[get_all_rrsets\(\)](#) (boto.route53.connection.Route53Connection method), 690
[get_all_scaling_process_types\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 180
[get_all_scheduled_actions\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 180
[get_all_security_groups\(\)](#) (boto.ec2.connection.EC2Connection method), 421
[get_all_server_certs\(\)](#) (boto.iam.connection.IAMConnection method), 580
[get_all_signing_certs\(\)](#) (boto.iam.connection.IAMConnection method), 580
[get_all_snapshots\(\)](#) (boto.ec2.connection.EC2Connection method), 421
[get_all_spot_instance_requests\(\)](#) (boto.ec2.connection.EC2Connection method), 422
[get_all_streaming_distributions\(\)](#) (boto.cloudfront.CloudFrontConnection method), 235
[get_all_subnets\(\)](#) (boto.vpc.VPCConnection method), 812
[get_all_subscriptions\(\)](#) (boto.sns.SNSConnection method), 751
[get_all_subscriptions_by_topic\(\)](#) (boto.sns.SNSConnection method), 751
[get_all_tags\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 180
[get_all_tags\(\)](#) (boto.ec2.connection.EC2Connection method), 422
[get_all_topics\(\)](#) (boto.sns.SNSConnection method), 751
[get_all_user_policies\(\)](#) (boto.iam.connection.IAMConnection method), 580
[get_all_users\(\)](#) (boto.iam.connection.IAMConnection method), 580
[get_all_versions\(\)](#) (boto.gs.bucket.Bucket method), 546
[get_all_versions\(\)](#) (boto.s3.bucket.Bucket method), 703
[get_all_volume_status\(\)](#) (boto.ec2.connection.EC2Connection method), 422
[get_all_volumes\(\)](#) (boto.ec2.connection.EC2Connection method), 423
[get_all_vpc_peering_connections\(\)](#) (boto.vpc.VPCConnection method), 813
[get_all_vpcs\(\)](#) (boto.vpc.VPCConnection method), 813
[get_all_vpn_connections\(\)](#) (boto.vpc.VPCConnection method), 814
[get_all_vpn_gateways\(\)](#) (boto.vpc.VPCConnection method), 814
[get_all_zones\(\)](#) (boto.ec2.connection.EC2Connection

method), 423

get_analysis_schemes() (boto.cloudsearch2.domain.DomainConnection method), 265

get_application() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_application_revision() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_archive_id() (boto.glacier.writer.Writer method), 537

get_as_params() (boto.mturk.layoutparam.LayoutParameter method), 605

get_as_params() (boto.mturk.layoutparam.LayoutParameter method), 605

get_as_params() (boto.mturk.price.Price method), 606

get_as_params() (boto.mturk.qualification.LocaleRequirement method), 606

get_as_params() (boto.mturk.qualification.Qualifications method), 607

get_as_params() (boto.mturk.qualification.Requirement method), 607

get_as_params() (boto.mturk.question.ExternalQuestion method), 608

get_as_params() (boto.mturk.question.HTMLQuestion method), 608

get_as_params() (boto.mturk.question.Overview method), 609

get_as_params() (boto.mturk.question.Question method), 609

get_as_xml() (boto.mturk.question.AnswerSpecification method), 607

get_as_xml() (boto.mturk.question.Application method), 607

get_as_xml() (boto.mturk.question.Constraint method), 608

get_as_xml() (boto.mturk.question.Constraints method), 608

get_as_xml() (boto.mturk.question.ExternalQuestion method), 608

get_as_xml() (boto.mturk.question.FileUploadAnswer method), 608

get_as_xml() (boto.mturk.question.FreeTextAnswer method), 608

get_as_xml() (boto.mturk.question.HTMLQuestion method), 608

get_as_xml() (boto.mturk.question.List method), 609

get_as_xml() (boto.mturk.question.NumberOfLinesSuggestion method), 609

get_as_xml() (boto.mturk.question.OrderedContent method), 609

get_as_xml() (boto.mturk.question.Overview method), 609

get_as_xml() (boto.mturk.question.Question method), 609

get_as_xml() (boto.mturk.question.QuestionContent method), 609

get_as_xml() (boto.mturk.question.QuestionForm method), 610

get_as_xml() (boto.mturk.question.SelectionAnswer method), 610

get_as_xml() (boto.mturk.question.XMLTemplate method), 611

get_assignment() (boto.mturk.connection.MTurkConnection method), 602

get_assignments() (boto.mturk.connection.MTurkConnection method), 603

get_attribute() (boto.ec2.instance.Instance method), 446

get_attributes() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504

get_attributes() (boto.mturk.question.Constraint method), 608

get_attributes() (boto.mturk.question.RegExConstraint method), 610

get_attributes() (boto.sdb.connection.SDBConnection method), 736

get_attributes() (boto.sdb.domain.Domain method), 739

get_attributes() (boto.sqs.queue.Queue method), 764

get_authorization_details() (boto.mws.connection.MWSConnection method), 613

get_availability_options() (boto.cloudsearch2.domain.Domain method), 265

get_aws_metadata() (in module boto.utils), 223

get_batch_prediction() (boto.machinelearning.layer1.MachineLearningConnection method), 844

get_billing_config() (boto.gs.bucket.Bucket method), 546

get_billing_configuration_with_xml() (boto.gs.bucket.Bucket method), 546

get_blob_bucket() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921

get_body() (boto.contrib.ymlmessage.YAMLMessage method), 321

get_body() (boto.sqs.message.RawMessage method), 763

get_body_encoded() (boto.sqs.message.RawMessage method), 763

get_bucket() (boto.file.connection.FileConnection method), 516

get_bucket() (boto.gs.connection.GSConnection method), 557

get_bucket() (boto.s3.connection.S3Connection method), 712

get_bucket_server() (boto.s3.connection.OrdinaryCallingFormat method), 711

get_bucket_server() (boto.s3.connection.SubdomainCallingFormat method), 714

get_bucket_server() (boto.s3.connection.VHostCallingFormat method), 714

get_canonical_user_id() (boto.gs.connection.GSConnection

method), 558

get_canonical_user_id() (boto.s3.connection.S3Connection method), 713

get_capture_details() (boto.mws.connection.MWSConnection method), 613

get_cartinfo_service_status() (boto.mws.connection.MWSConnection method), 613

get_carts() (boto.mws.connection.MWSConnection method), 613

get_change() (boto.route53.connection.Route53Connection method), 691

get_checker_ip_ranges() (boto.route53.connection.Route53Connection method), 691

get_choices() (boto.sdb.db.property.Property method), 924

get_cname() (boto.route53.zone.Zone method), 696

get_competitive_pricing_for_asin() (boto.mws.connection.MWSConnection method), 613

get_competitive_pricing_for_sku() (boto.mws.connection.MWSConnection method), 613

get_complex_tuple() (boto.ec2.elb.listener.Listener method), 502

get_config() (boto.cloudhsm.layer1.CloudHSMConnection method), 245

get_console_output() (boto.ec2.connection.EC2Connection method), 423

get_console_output() (boto.ec2.instance.Instance method), 446

get_contents_as_string() (boto.file.key.Key method), 516

get_contents_as_string() (boto.gs.key.Key method), 563

get_contents_as_string() (boto.s3.key.Key method), 719

get_contents_to_file() (boto.file.key.Key method), 517

get_contents_to_file() (boto.gs.key.Key method), 564

get_contents_to_file() (boto.s3.key.Key method), 720

get_contents_to_filename() (boto.gs.key.Key method), 564

get_contents_to_filename() (boto.s3.key.Key method), 720

get_cors() (boto.gs.bucket.Bucket method), 547

get_cors() (boto.s3.bucket.Bucket method), 703

get_cors_xml() (boto.gs.bucket.Bucket method), 547

get_cors_xml() (boto.s3.bucket.Bucket method), 704

get_credential_report() (boto.iam.connection.IAMConnection method), 581

get_cur_file_size() (in module boto.s3.resumable_download_handler), 730

get_customerinfo_service_status() (boto.mws.connection.MWSConnection method), 613

get_customers_for_customer_id() (boto.mws.connection.MWSConnection method), 613

get_data_source() (boto.machinelearning.layer1.MachineLearningConnection method), 844

get_dead_letter_source_queues() (boto.sqs.connection.SQSConnection method), 757

get_debt_balance() (boto.fps.connection.FPSConnection method), 518

get_def_acl() (boto.gs.bucket.Bucket method), 547

get_deployment() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_deployment_config() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_deployment_group() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_deployment_instance() (boto.codedeploy.layer1.CodeDeployConnection method), 299

get_distribution() (boto.cloudfront.distribution.DistributionSummary method), 238

get_distribution() (boto.cloudfront.distribution.StreamingDistributionSummary method), 239

get_distribution() (boto.cloudfront.invalidation.InvalidationSummary method), 242

get_distribution_config() (boto.cloudfront.CloudFrontConnection method), 235

get_distribution_info() (boto.cloudfront.CloudFrontConnection method), 235

get_document_service() (boto.cloudsearch.domain.Domain method), 249

get_document_service() (boto.cloudsearch2.domain.Domain method), 265

get_domain() (boto.sdb.connection.SDBConnection method), 736

get_domain_and_name() (boto.sdb.connection.SDBConnection method), 736

get_domain_detail() (boto.route53.domains.layer1.Route53DomainsConnection method), 911

get_dynamodb_type() (in module boto.dynamodb.types), 348

get_encryption_config() (boto.gs.bucket.Bucket method), 547

get_endpoint_attributes() (boto.sns.SNSConnection method), 751

get_etag() (boto.cloudfront.CloudFrontConnection method), 235

get_evaluation() (boto.machinelearning.layer1.MachineLearningConnection method), 844

get_event_source() (boto.awslambda.layer1.AWSLambdaConnection method), 189

get_expressions() (boto.cloudsearch2.domain.Domain

method), 265

get_federation_token() (boto.sts.STSConnection method), 774

get_feed_submission_count() (boto.mws.connection.MWSConnection method), 613

get_feed_submission_list() (boto.mws.connection.MWSConnection method), 613

get_feed_submission_list_by_next_token() (boto.mws.connection.MWSConnection method), 614

get_feed_submission_result() (boto.mws.connection.MWSConnection method), 614

get_file() (boto.file.key.Key method), 517

get_file() (boto.gs.key.Key method), 565

get_file() (boto.manage.cmdshell.LocalClient method), 597

get_file() (boto.manage.cmdshell.SSHClient method), 598

get_file() (boto.s3.key.Key method), 721

get_file() (boto.s3.resumable_download_handler.ResumableDownloadHandler method), 729

get_file() (boto.services.service.Service method), 744

get_file_upload_url() (boto.mturk.connection.MTurkConnection method), 603

get_fulfillment_order() (boto.mws.connection.MWSConnection method), 614

get_fulfillment_preview() (boto.mws.connection.MWSConnection method), 614

get_function() (boto.awslambda.layer1.AWSLambdaConnection method), 189

get_function_configuration() (boto.awslambda.layer1.AWSLambdaConnection method), 189

get_group() (boto.iam.connection.IAMConnection method), 581

get_group_policy() (boto.iam.connection.IAMConnection method), 581

get_groups_for_user() (boto.iam.connection.IAMConnection method), 581

get_help() (boto.mturk.connection.MTurkConnection method), 603

get_hit() (boto.mturk.connection.MTurkConnection method), 603

get_hosted_zone() (boto.route53.connection.Route53Connection method), 691

get_hosted_zone_by_name() (boto.route53.connection.Route53Connection method), 691

get_hostname_suggestion() (boto.opsworks.layer1.OpsWorksConnection method), 864

get_http_connection() (boto.connection.AWSAuthConnection method), 215

get_http_connection() (boto.connection.ConnectionPool method), 216

get_http_connection() (boto.gs.connection.GSConnection method), 558

get_id() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 308

get_identity_dkim_attributes() (boto.ses.connection.SESConnection method), 746

get_identity_pool_configuration() (boto.cognito.sync.layer1.CognitoSyncConnection method), 313

get_identity_verification_attributes() (boto.ses.connection.SESConnection method), 746

get_image() (boto.ec2.connection.EC2Connection method), 423

get_image_attribute() (boto.ec2.connection.EC2Connection method), 424

get_image_device_status() (boto.mws.connection.MWSConnection method), 614

get_index_fields() (boto.cloudsearch.domain.Domain method), 249

get_index_fields() (boto.cloudsearch2.domain.Domain method), 265

get_inner_content() (boto.mturk.question.Application method), 607

get_inner_content() (boto.mturk.question.Flash method), 608

get_inner_content() (boto.mturk.question.JavaApplet method), 608

get_instance() (boto.pyami.config.Config method), 631

get_instance_attribute() (boto.ec2.connection.EC2Connection method), 424

get_instance_health() (boto.ec2.elb.loadbalancer.LoadBalancer method), 504

get_instance_identity() (in module boto.utils), 223

get_instance_metadata() (in module boto.utils), 223

get_instance_profile() (boto.iam.connection.IAMConnection method), 581

get_instance_type() (boto.ec2.buyreservation.BuyReservation method), 405

get_instance_userdata() (in module boto.utils), 223

get_invalidation_request() (boto.cloudfront.invalidation.InvalidationSummary method), 242

get_invalidation_requests() (boto.cloudfront.CloudFrontConnection method), 235

get_inventory_service_status()

- (boto.mws.connection.MWSSConnection method), 614
- get_item() (boto.dynamodb.layer1.Layer1 method), 329
- get_item() (boto.dynamodb.layer2.Layer2 method), 333
- get_item() (boto.dynamodb.table.Table method), 339
- get_item() (boto.dynamodb2.layer1.DynamoDBConnection method), 380
- get_item() (boto.dynamodb2.table.Table method), 362
- get_item() (boto.sdb.domain.Domain method), 739
- get_job() (boto.glacier.vault.Vault method), 534
- get_job_output() (boto.glacier.layer1.Layer1 method), 525
- get_kernel() (boto.ec2.image.Image method), 441
- get_key() (boto.file.bucket.Bucket method), 515
- get_key() (boto.gs.bucket.Bucket method), 547
- get_key() (boto.s3.bucket.Bucket method), 704
- get_key_fields() (boto.dynamodb2.table.Table method), 363
- get_key_name() (boto.services.submit.Submitter method), 745
- get_key_pair() (boto.ec2.connection.EC2Connection method), 424
- get_key_policy() (boto.kms.layer1.KMSConnection method), 825
- get_key_rotation_status() (boto.kms.layer1.KMSConnection method), 826
- get_key_value() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
- get_keys() (boto.dynamodb2.items.Item method), 354
- get_keywords_as_string() (boto.mturk.connection.MTurkConnection static method), 603
- get_last_updated_time_for_recommendations() (boto.mws.connection.MWSSConnection method), 614
- get_launch_permissions() (boto.ec2.image.Image method), 441
- get_lb_attribute() (boto.ec2.elb.ELBConnection method), 500
- get_lifecycle_config() (boto.gs.bucket.Bucket method), 547
- get_lifecycle_config() (boto.s3.bucket.Bucket method), 704
- get_list() (boto.connection.AWSQueryConnection method), 216
- get_list_health_checks() (boto.route53.connection.Route53Connection method), 691
- get_location() (boto.gs.bucket.Bucket method), 548
- get_location() (boto.s3.bucket.Bucket method), 704
- get_log_events() (boto.logs.layer1.CloudWatchLogsConnection method), 831
- get_log_file() (boto.rds.RDSConnection method), 643
- get_logging_config() (boto.gs.bucket.Bucket method), 548
- get_logging_config_with_xml() (boto.gs.bucket.Bucket method), 548
- get_logging_status() (boto.gs.bucket.Bucket method), 548
- get_logging_status() (boto.s3.bucket.Bucket method), 704
- get_login_profiles() (boto.iam.connection.IAMConnection method), 581
- get_lowest_offer_listings_for_asin() (boto.mws.connection.MWSSConnection method), 614
- get_lowest_offer_listings_for_sku() (boto.mws.connection.MWSSConnection method), 614
- get_manager() (in module boto.sdb.db.manager), 919
- get_matching_product() (boto.mws.connection.MWSSConnection method), 614
- get_matching_product_for_id() (boto.mws.connection.MWSSConnection method), 615
- get_md5_from_hexdigest() (boto.gs.key.Key method), 565
- get_md5_from_hexdigest() (boto.s3.key.Key method), 721
- get_messages() (boto.sqs.queue.Queue method), 765
- get_metadata() (boto.gs.key.Key method), 565
- get_metadata() (boto.s3.key.Key method), 721
- get_metadata() (boto.sdb.domain.Domain method), 740
- get_metric_statistics() (boto.ec2.cloudwatch.CloudWatchConnection method), 290
- get_ml_model() (boto.machinelearning.layer1.MachineLearningConnection method), 844
- get_mx() (boto.route53.zone.Zone method), 696
- get_my_price_for_asin() (boto.mws.connection.MWSSConnection method), 615
- get_my_price_for_sku() (boto.mws.connection.MWSSConnection method), 615
- get_nameservers() (boto.route53.zone.Zone method), 696
- get_next_token() (boto.sdb.db.query.Query method), 926
- get_num_hits() (boto.cloudsearch.search.SearchConnection method), 259
- get_num_hits() (boto.cloudsearch2.search.SearchConnection method), 274
- get_oai_value() (in module boto.cloudfront.origin), 240
- get_obj() (boto.services.servicedef.ServiceDef method), 744
- get_object() (boto.connection.AWSQueryConnection method), 216
- get_object() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
- get_object_from_id() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
- get_objects() (boto.cloudfront.distribution.Distribution

method), 236

get_offamazonpayments_service_status()
(boto.mws.connection.MWSConnection method), 615

get_only_instances() (boto.ec2.connection.EC2Connection method), 424

get_open_id_token() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 308

get_open_id_token_for_developer_identity()
(boto.cognito.identity.layer1.CognitoIdentityConnection method), 308

get_operation_detail() (boto.route53.domains.layer1.Route53DomainsConnection method), 911

get_order() (boto.mws.connection.MWSConnection method), 615

get_order_reference_details()
(boto.mws.connection.MWSConnection method), 615

get_orders_service_status()
(boto.mws.connection.MWSConnection method), 615

get_origin_access_identity()
(boto.cloudfront.identity.OriginAccessIdentitySummary method), 241

get_origin_access_identity_config()
(boto.cloudfront.CloudFrontConnection method), 235

get_origin_access_identity_info()
(boto.cloudfront.CloudFrontConnection method), 235

get_outbound_service_status()
(boto.mws.connection.MWSConnection method), 615

get_output() (boto.glacier.job.Job method), 537

get_outstanding_debt_balance()
(boto.fps.connection.FPSConnection method), 519

get_package_tracking_details()
(boto.mws.connection.MWSConnection method), 615

get_params() (boto.ec2.connection.EC2Connection method), 425

get_params() (boto.rds.parametergroup.ParameterGroup method), 654

get_password_data() (boto.ec2.connection.EC2Connection method), 425

get_path() (boto.connection.AWSAuthConnection method), 215

get_path() (boto.gs.connection.GSConnection method), 558

get_payment_instruction()
(boto.fps.connection.FPSConnection method), 519

get_permissions() (boto.ec2.snapshot.Snapshot method), 458

get_pipeline_definition() (boto.datapipeline.layer1.DataPipelineConnection method), 323

get_platform_application_attributes()
(boto.sns.SNSConnection method), 751

get_policy() (boto.cloudformation.stack.Stack method), 323

get_policy() (boto.gs.bucket.Bucket method), 548

get_policy() (boto.iam.connection.IAMConnection method), 581

get_policy() (boto.s3.bucket.Bucket method), 704

get_policy() (boto.iam.connection.IAMConnection method), 581

get_prepaid_balance() (boto.fps.connection.FPSConnection method), 519

get_price_as_price() (boto.mturk.connection.MTurkConnection static method), 603

get_product_categories_for_asin()
(boto.mws.connection.MWSConnection method), 615

get_product_categories_for_sku()
(boto.mws.connection.MWSConnection method), 615

get_products_service_status()
(boto.mws.connection.MWSConnection method), 616

get_property() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921

get_proxy_auth_header()
(boto.connection.AWSAuthConnection method), 215

get_proxy_auth_header()
(boto.gs.connection.GSConnection method), 558

get_proxy_url_with_auth()
(boto.connection.AWSAuthConnection method), 215

get_proxy_url_with_auth()
(boto.gs.connection.GSConnection method), 558

get_qualification_requests()
(boto.mturk.connection.MTurkConnection method), 603

get_qualification_score()
(boto.mturk.connection.MTurkConnection method), 603

get_qualification_type() (boto.mturk.connection.MTurkConnection method), 603

get_qualifications_for_qualification_type()
(boto.mturk.connection.MTurkConnection method), 603

get_quantity() (boto.ec2.buyreservation.BuyReservation method), 405

get_query() (boto.sdb.db.query.Query method), 926

get_queue() (boto.sqs.connection.SQSConnection method), 757
 get_queue_attributes() (boto.sqs.connection.SQSConnection method), 757
 get_ramdisk() (boto.ec2.image.Image method), 441
 get_rank_expressions() (boto.cloudsearch.domain.Domain method), 249
 get_raw_item() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 get_raw_keys() (boto.dynamodb2.items.Item method), 354
 get_recipient_verification_status() (boto.fps.connection.FPSConnection method), 519
 get_recommendations_service_status() (boto.mws.connection.MWSConnection method), 616
 get_records() (boto.kinesis.layer1.KinesisConnection method), 590
 get_records() (boto.route53.zone.Zone method), 696
 get_redirect() (boto.gs.key.Key method), 565
 get_redirect() (boto.s3.key.Key method), 721
 get_refund_details() (boto.mws.connection.MWSConnection method), 616
 get_region() (boto.ec2.buyreservation.BuyReservation method), 405
 get_region() (in module boto.ec2), 403
 get_report() (boto.mws.connection.MWSConnection method), 616
 get_report_count() (boto.mws.connection.MWSConnection method), 616
 get_report_list() (boto.mws.connection.MWSConnection method), 616
 get_report_list_by_next_token() (boto.mws.connection.MWSConnection method), 616
 get_report_request_count() (boto.mws.connection.MWSConnection method), 616
 get_report_request_list() (boto.mws.connection.MWSConnection method), 616
 get_report_request_list_by_next_token() (boto.mws.connection.MWSConnection method), 616
 get_report_schedule_count() (boto.mws.connection.MWSConnection method), 617
 get_report_schedule_list() (boto.mws.connection.MWSConnection method), 617
 get_report_schedule_list_by_next_token() (boto.mws.connection.MWSConnection method), 617
 get_request_payment() (boto.gs.bucket.Bucket method), 548
 get_request_payment() (boto.s3.bucket.Bucket method), 704
 get_resource_config_history() (boto.configservice.layer1.ConfigServiceConnection method), 318
 get_response() (boto.cloudsearch.layer1.Layer1 method), 254
 get_response() (boto.ecs.ECSConnection method), 470
 get_response() (boto.iam.connection.IAMConnection method), 582
 get_results() (boto.services.result.ResultProcessor method), 743
 get_results_from_bucket() (boto.services.result.ResultProcessor method), 743
 get_results_from_domain() (boto.services.result.ResultProcessor method), 743
 get_results_from_queue() (boto.services.result.ResultProcessor method), 743
 get_reviewable_hits() (boto.mturk.connection.MTurkConnection method), 603
 get_role() (boto.iam.connection.IAMConnection method), 582
 get_role_policy() (boto.iam.connection.IAMConnection method), 582
 get_s3_connection() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 get_saml_provider() (boto.iam.connection.IAMConnection method), 582
 get_scaling_options() (boto.cloudsearch2.domain.Domain method), 265
 get_sdf() (boto.cloudsearch.document.DocumentServiceConnection method), 262
 get_sdf() (boto.cloudsearch2.document.DocumentServiceConnection method), 277
 get_search_service() (boto.cloudsearch.domain.Domain method), 249
 get_search_service() (boto.cloudsearch2.domain.Domain method), 265
 get_send_quota() (boto.ses.connection.SESConnection method), 746
 get_send_statistics() (boto.ses.connection.SESConnection method), 746
 get_server_certificate() (boto.iam.connection.IAMConnection method), 582
 get_service_status() (boto.mws.connection.MWSConnection method), 617
 get_session_token() (boto.sts.STSConnection method), 775
 get_shard_iterator() (boto.kinesis.layer1.KinesisConnection method), 591

[get_signin_url\(\)](#) (boto.iam.connection.IAMConnection method), 582
[get_snapshot_attribute\(\)](#) (boto.ec2.connection.EC2Connection method), 425
[get_spot_datafeed_subscription\(\)](#) (boto.ec2.connection.EC2Connection method), 425
[get_spot_price_history\(\)](#) (boto.ec2.connection.EC2Connection method), 425
[get_stack_policy\(\)](#) (boto.cloudformation.connection.CloudFormationConnection method), 229
[get_status\(\)](#) (boto.connection.AWSQueryConnection method), 216
[get_stemming\(\)](#) (boto.cloudsearch.domain.Domain method), 249
[get_stopwords\(\)](#) (boto.cloudsearch.domain.Domain method), 249
[get_storage_class\(\)](#) (boto.gs.bucket.Bucket method), 548
[get_streaming_distribution_config\(\)](#) (boto.cloudfront.CloudFrontConnection method), 235
[get_streaming_distribution_info\(\)](#) (boto.cloudfront.CloudFrontConnection method), 235
[get_subresource\(\)](#) (boto.gs.bucket.Bucket method), 548
[get_subresource\(\)](#) (boto.s3.bucket.Bucket method), 704
[get_subscription\(\)](#) (boto.mws.connection.MWSConnection method), 617
[get_subscription_details\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_subscriptions_service_status\(\)](#) (boto.mws.connection.MWSConnection method), 617
[get_synonyms\(\)](#) (boto.cloudsearch.domain.Domain method), 249
[get_table\(\)](#) (boto.dynamodb.layer2.Layer2 method), 333
[get_tags\(\)](#) (boto.gs.bucket.Bucket method), 549
[get_tags\(\)](#) (boto.s3.bucket.Bucket method), 705
[get_template\(\)](#) (boto.cloudformation.connection.CloudFormationConnection method), 229
[get_template\(\)](#) (boto.cloudformation.stack.Stack method), 233
[get_termination_policies\(\)](#) (boto.ec2.autoscale.AutoScaleConnection method), 180
[get_timeout\(\)](#) (boto.sqs.queue.Queue method), 765
[get_token_by_caller\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_token_usage\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_tokens\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_topic_attributes\(\)](#) (boto.sns.SNSConnection method), 751
[get_torrent_file\(\)](#) (boto.gs.key.Key method), 566
[get_torrent_file\(\)](#) (boto.s3.key.Key method), 721
[get_total_prepaid_liability\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_tracker_uri\(\)](#) (boto.gs.resumable_upload_handler.ResumableUploadHandler method), 572
[get_trail_status\(\)](#) (boto.cloudtrail.layer1.CloudTrailConnection method), 285
[get_transaction\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_transaction_status\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_transactions_for_subscription\(\)](#) (boto.fps.connection.FPSConnection method), 519
[get_ts\(\)](#) (in module boto.utils), 223
[get_tuple\(\)](#) (boto.ec2.elb.listener.Listener method), 502
[get_upload_id\(\)](#) (boto.gs.resumable_upload_handler.ResumableUploadHandler method), 572
[get_usage\(\)](#) (boto.sdb.connection.SDBConnection method), 736
[get_user\(\)](#) (boto.iam.connection.IAMConnection method), 582
[get_user\(\)](#) (boto.pyami.config.Config method), 631
[get_user_policy\(\)](#) (boto.iam.connection.IAMConnection method), 582
[get_utf8_value\(\)](#) (boto.connection.AWSQueryConnection method), 216
[get_utf8_value\(\)](#) (in module boto.utils), 223
[get_value\(\)](#) (boto.pyami.config.Config method), 631
[get_value\(\)](#) (boto.rds.parametergroup.Parameter method), 653
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.CalculatedProperty method), 922
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.DateProperty method), 922
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.DateTimeProperty method), 922
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.PasswordProperty method), 924
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.Property method), 924
[get_value_for_datastore\(\)](#) (boto.sdb.db.property.S3KeyProperty method), 925
[get_vault\(\)](#) (boto.glacier.layer2.Layer2 method), 533
[get_vault_notifications\(\)](#) (boto.glacier.layer1.Layer1 method), 525

- get_versioning_status() (boto.gs.bucket.Bucket method), 549
- get_versioning_status() (boto.s3.bucket.Bucket method), 705
- get_volume_attribute() (boto.ec2.connection.EC2Connection method), 426
- get_website_configuration() (boto.gs.bucket.Bucket method), 549
- get_website_configuration() (boto.s3.bucket.Bucket method), 705
- get_website_configuration_obj() (boto.gs.bucket.Bucket method), 549
- get_website_configuration_obj() (boto.s3.bucket.Bucket method), 705
- get_website_configuration_with_xml() (boto.gs.bucket.Bucket method), 549
- get_website_configuration_with_xml() (boto.s3.bucket.Bucket method), 705
- get_website_configuration_xml() (boto.gs.bucket.Bucket method), 550
- get_website_configuration_xml() (boto.s3.bucket.Bucket method), 705
- get_website_endpoint() (boto.gs.bucket.Bucket method), 550
- get_website_endpoint() (boto.s3.bucket.Bucket method), 705
- get_workflow_execution_history() (boto.swf.layer1.Layer1 method), 780
- get_xml_acl() (boto.gs.bucket.Bucket method), 550
- get_xml_acl() (boto.gs.key.Key method), 566
- get_xml_acl() (boto.s3.bucket.Bucket method), 706
- get_xml_acl() (boto.s3.key.Key method), 722
- get_xml_tags() (boto.gs.bucket.Bucket method), 550
- get_xml_tags() (boto.s3.bucket.Bucket method), 706
- get_zone() (boto.ec2.buyreservation.BuyReservation method), 405
- get_zone() (boto.route53.connection.Route53Connection method), 691
- get_zones() (boto.route53.connection.Route53Connection method), 691
- GetAuthorizationDetailsResult (class in boto.mws.response), 624
- getbool() (boto.pyami.config.Config method), 631
- getbool() (boto.services.servicedef.ServiceDef method), 744
- GetCaptureDetailsResult (class in boto.mws.response), 624
- GetCartsResult (class in boto.mws.response), 624
- GetCompetitivePricingForASINResponse (class in boto.mws.response), 624
- GetCompetitivePricingForSKUResponse (class in boto.mws.response), 624
- GetCustomersForCustomerIdResult (class in boto.mws.response), 624
- GetFeedSubmissionCountResult (class in boto.mws.response), 624
- GetFeedSubmissionListResult (class in boto.mws.response), 624
- getfloat() (boto.pyami.config.Config method), 631
- GetFulfillmentOrderResult (class in boto.mws.response), 624
- GetFulfillmentPreviewResult (class in boto.mws.response), 625
- getint() (boto.pyami.config.Config method), 631
- getint() (boto.services.servicedef.ServiceDef method), 744
- getint_user() (boto.pyami.config.Config method), 631
- GetLowestOfferListingsForASINResponse (class in boto.mws.response), 625
- GetLowestOfferListingsForSKUResponse (class in boto.mws.response), 625
- GetMatchingProductForIdResponse (class in boto.mws.response), 625
- GetMatchingProductForIdResult (class in boto.mws.response), 625
- GetMatchingProductResponse (class in boto.mws.response), 625
- GetMyPriceForASINResponse (class in boto.mws.response), 625
- GetMyPriceForSKUResponse (class in boto.mws.response), 625
- GetOrderReferenceDetailsResult (class in boto.mws.response), 625
- GetOrderResult (class in boto.mws.response), 625
- getOutput() (boto.utils.ShellCommand method), 222
- GetPackageTrackingDetailsResult (class in boto.mws.response), 625
- GetProductCategoriesForASINResult (class in boto.mws.response), 625
- GetProductCategoriesForSKUResult (class in boto.mws.response), 625
- GetProductCategoriesResult (class in boto.mws.response), 625
- GetRefundDetails (class in boto.mws.response), 625
- GetReportListResult (class in boto.mws.response), 625
- GetReportRequestListResult (class in boto.mws.response), 625
- GetReportScheduleListResult (class in boto.mws.response), 625
- GetServiceStatusResult (class in boto.mws.response), 626
- getStatus() (boto.utils.ShellCommand method), 222
- GetSubscriptionResult (class in boto.mws.response), 626
- GiftWrapPrice (boto.mws.response.OrderItem attribute), 628
- GiftWrapTax (boto.mws.response.OrderItem attribute), 628
- GlobalAllIndex (class in boto.dynamodb2.fields), 350

GlobalBaseIndexField (class in boto.dynamodb2.fields), 350

GlobalIncludeIndex (class in boto.dynamodb2.fields), 351

GlobalKeysOnlyIndex (class in boto.dynamodb2.fields), 351

Grant (class in boto.s3.acl), 697

grant_bonus() (boto.mturk.connection.MTurkConnection method), 603

grant_qualification() (boto.mturk.connection.MTurkConnection method), 603

Group (class in boto.ec2.group), 441

GroupList (class in boto.ec2.launchspecification), 451

GroupOrCIDR (class in boto.ec2.securitygroup), 456

gs_access_key_id (boto.connection.AWSAuthConnection attribute), 215

gs_access_key_id (boto.gs.connection.GSConnection attribute), 558

gs_secret_access_key (boto.connection.AWSAuthConnection attribute), 215

gs_secret_access_key (boto.gs.connection.GSConnection attribute), 558

GSConnection (class in boto.gs.connection), 556

GSCopyError, 218

GSCreateError, 218

GSDataError, 218

GSPermissionsError, 218

GSResponseError, 218

guess_mime_type() (in module boto.utils), 223

H

HadoopStep (class in boto.emr.emrobject), 514

handle_addl_headers() (boto.gs.key.Key method), 566

handle_addl_headers() (boto.s3.key.Key method), 722

handle_bool() (in module boto.cloudsearch.domain), 250

handle_bool() (in module boto.cloudsearch2.domain), 266

handle_encryption_headers() (boto.gs.key.Key method), 566

handle_encryption_headers() (boto.s3.key.Key method), 722

handle_proxy() (boto.connection.AWSAuthConnection method), 215

handle_proxy() (boto.gs.connection.GSConnection method), 558

handle_request_data() (boto.utils.RequestHook method), 222

handle_restore_headers() (boto.gs.key.Key method), 566

handle_restore_headers() (boto.s3.key.Key method), 722

handle_resumable_upload_exception() (boto.gs.resumable_upload_handler.ResumableUploadHandler method), 572

handle_storage_class_header() (boto.gs.key.Key method), 566

handle_storage_class_header() (boto.s3.key.Key method), 722

handle_unprocessed() (boto.dynamodb2.table.BatchTable method), 357

handle_version_headers() (boto.gs.key.Key method), 566

handle_version_headers() (boto.s3.key.Key method), 722

has_id_or_name() (boto.sdb.db.key.Key method), 919

has_item() (boto.dynamodb.table.Table method), 339

has_item() (boto.dynamodb2.table.Table method), 363

has_key() (boto.sqs.message.MHMessage method), 762

has_option() (boto.pyami.config.Config method), 631

has_option() (boto.services.servicedef.ServiceDef method), 744

hash_key (boto.dynamodb.item.Item attribute), 345

hash_key_name (boto.dynamodb.item.Item attribute), 345

hash_key_name (boto.dynamodb.schema.Schema attribute), 344

hash_key_type (boto.dynamodb.schema.Schema attribute), 344

hashfunc() (boto.utils.Password method), 222

HashKey (class in boto.dynamodb2.fields), 352

head_bucket() (boto.gs.connection.GSConnection method), 558

head_bucket() (boto.s3.connection.S3Connection method), 713

HealthCheck (class in boto.ec2.elb.healthcheck), 501

HealthCheck (class in boto.route53.healthcheck), 692

HealthCheckBody (boto.route53.record.Record attribute), 693

heartbeat() (boto.swf.layer2.ActivityWorker method), 793

history() (boto.swf.layer2.WorkflowExecution method), 797

HIT (class in boto.mturk.connection), 600

HITTypeId (class in boto.mturk.connection), 601

HiveBase (class in boto.emr.step), 510

HiveStep (class in boto.emr.step), 510

host_is_ipv6() (in module boto.utils), 223

HostConnectionPool (class in boto.connection), 217

HostedZone (class in boto.route53.hostedzone), 693

HostRequiredError, 710

HsmClientCertificateAlreadyExists, 684

HsmClientCertificateNotFound, 684

HsmClientCertificateQuotaExceeded, 684

HsmConfigurationAlreadyExists, 684

HsmConfigurationNotFound, 684

HsmConfigurationQuotaExceeded, 684

HTMLQuestion (class in boto.mturk.question), 608

HTTPRequest (class in boto.connection), 216

HTTPResponse (class in boto.connection), 217

I

IAMConnection (class in boto.iam.connection), 574

- IAMRegionInfo (class in boto.iam), 573
- id (boto.cloudsearch.domain.Domain attribute), 249
- id (boto.cloudsearch2.domain.Domain attribute), 265
- id (boto.sqs.queue.Queue attribute), 765
- id() (boto.sdb.db.key.Key method), 919
- id_or_name() (boto.sdb.db.key.Key method), 919
- IdempotentParameterMismatchException, 846
- Identifiers (boto.mws.response.Product attribute), 628
- Identifiers (boto.mws.response.VariationRelationship attribute), 630
- Image (class in boto.ec2.image), 441
- Image (class in boto.mws.response), 626
- ImageAttribute (class in boto.ec2.image), 444
- import_key_pair() (boto.ec2.connection.EC2Connection method), 426
- InboundShipmentPlans (boto.mws.response.CreateInboundShipmentPlan attribute), 623
- IncludeIndex (class in boto.dynamodb2.fields), 352
- IncompatibleOrderableOptions, 684
- IncompatibleVersionException, 495
- index_documents() (boto.cloudsearch.domain.Domain method), 249
- index_documents() (boto.cloudsearch.layer1.Layer1 method), 254
- index_documents() (boto.cloudsearch2.domain.Domain method), 265
- index_documents() (boto.cloudsearch2.layer1.CloudSearchConnection method), 270
- IndexFieldStatus (class in boto.cloudsearch.optionstatus), 257
- IndexFieldStatus (class in boto.cloudsearch2.optionstatus), 272
- init_logging() (in module boto), 213
- initiate_job() (boto.glacier.layer1.Layer1 method), 526
- initiate_multipart_upload() (boto.glacier.layer1.Layer1 method), 527
- initiate_multipart_upload() (boto.gs.bucket.Bucket method), 550
- initiate_multipart_upload() (boto.s3.bucket.Bucket method), 706
- install() (boto.pyami.installers.Installer method), 631
- install() (boto.pyami.installers.ubuntu.apache.Apache method), 632
- install() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- install() (boto.pyami.installers.ubuntu.mysql.MySQL method), 633
- install() (boto.pyami.installers.ubuntu.trac.Trac method), 633
- install_payment_instruction() (boto.fps.connection.FPSConnection method), 520
- Installer (class in boto.pyami.installers), 631
- Installer (class in boto.pyami.installers.ubuntu.installer), 632
- InstallHiveName (boto.emr.step.InstallHiveStep attribute), 511
- InstallHiveStep (class in boto.emr.step), 511
- InstallPigName (boto.emr.step.InstallPigStep attribute), 511
- InstallPigStep (class in boto.emr.step), 511
- Instance (class in boto.beanstalk.response), 203
- Instance (class in boto.ec2.autoscale.instance), 184
- Instance (class in boto.ec2.instance), 444
- InstanceAttribute (class in boto.ec2.instance), 448
- InstanceCount (class in boto.ec2.reservedinstance), 454
- InstanceDoesNotExistException, 304
- InstanceGroup (class in boto.emr.emrobject), 514
- InstanceGroupInfo (class in boto.emr.emrobject), 514
- InstanceGroupInfoResult (class in boto.emr.emrobject), 514
- InstanceIdRequiredException, 304
- InstanceInfo (class in boto.ec2.instanceinfo), 449
- InstanceInfo (class in boto.emr.emrobject), 514
- InstanceList (class in boto.emr.emrobject), 514
- InstanceMonitoring (class in boto.ec2.autoscale.launchconfig), 184
- InstancePlacement (class in boto.ec2.instance), 448
- InstanceQuotaExceeded, 875
- instances() (boto.ec2.securitygroup.SecurityGroup method), 457
- InstanceState (class in boto.ec2.elb.instancestate), 501
- InstanceState (class in boto.ec2.instance), 448
- InstanceState (class in boto.ec2.instancestatus), 450
- InstanceStateSet (class in boto.ec2.instancestatus), 450
- INSUFFICIENT_DATA (boto.ec2.cloudwatch.alarm.MetricAlarm attribute), 295
- InsufficientClusterCapacity, 684
- InsufficientClusterCapacityFault, 684
- InsufficientDBInstanceCapacity, 875
- InsufficientDeliveryPolicyException, 319
- InsufficientS3BucketPolicy, 684
- InsufficientS3BucketPolicyException, 286
- InsufficientSnsTopicPolicyException, 286
- IntegerProperty (class in boto.sdb.db.property), 922
- InternalErrorException, 286, 311, 316
- InternalServerError, 402, 932
- InternalServerException, 846
- InternalServiceError, 327
- InternalServiceException, 495
- InternetGateway (class in boto.vpc.internetgateway), 817
- InternetGatewayAttachment (class in boto.vpc.internetgateway), 818
- InvalidAclError, 218
- InvalidAddress, 621
- InvalidAliasNameException, 828
- InvalidApplicationNameException, 305
- InvalidArgumentException, 596
- InvalidArnException, 828

invalidation_request_status()
 (boto.cloudfront.CloudFrontConnection
 method), 235
 InvalidationBatch (class in boto.cloudfront.invalidation),
 241
 InvalidationListResultSet (class
 in boto.cloudfront.invalidation), 241
 InvalidationSummary (class
 in boto.cloudfront.invalidation), 242
 InvalidAutoScalingGroupException, 305
 InvalidBucketNameFilterException, 305
 InvalidCiphertextException, 828
 InvalidCloudWatchLogsLogGroupArnException, 287
 InvalidCloudWatchLogsRoleArnException, 287
 InvalidClusterParameterGroupState, 685
 InvalidClusterParameterGroupStateFault, 685
 InvalidClusterSecurityGroupState, 685
 InvalidClusterSecurityGroupStateFault, 685
 InvalidClusterSnapshotState, 685
 InvalidClusterSnapshotStateFault, 685
 InvalidClusterState, 685
 InvalidClusterStateFault, 685
 InvalidClusterSubnetGroupState, 685
 InvalidClusterSubnetGroupStateFault, 685
 InvalidClusterSubnetState, 685
 InvalidClusterSubnetStateFault, 685
 InvalidConfigurationException, 316
 InvalidConfigurationRecorderNameException, 319
 InvalidCorsError, 218
 InvalidDBInstanceState, 875
 InvalidDBParameterGroupState, 875
 InvalidDBSecurityGroupState, 875
 InvalidDBSnapshotState, 875
 InvalidDBSubnetGroupState, 875
 InvalidDBSubnetState, 875
 InvalidDeliveryChannelNameException, 319
 InvalidDeployedStateFilterException, 305
 InvalidDeploymentConfigNameException, 305
 InvalidDeploymentGroupNameException, 305
 InvalidDeploymentIdException, 305
 InvalidDeploymentStatusException, 305
 InvalidEC2TagException, 305
 InvalidElasticIp, 685
 InvalidEncryptionConfigError, 218
 InvalidEventSubscriptionState, 875
 InvalidGrantTokenException, 828
 InvalidHsmClientCertificateState, 685
 InvalidHsmConfigurationState, 686
 InvalidInput, 918
 InvalidInputException, 846
 InvalidInstanceMetadataError, 218
 InvalidInstanceStateException, 305
 InvalidKeyPrefixFilterException, 305
 InvalidKeyUsageException, 828
 InvalidLifecycleConfigError, 218
 InvalidLimitException, 319
 InvalidLookupAttributesException, 287
 InvalidMarkerException, 828
 InvalidMaxResultsException, 287
 InvalidMinimumHealthyHostValueException, 306
 InvalidNextTokenException, 287, 306, 319
 InvalidOperationException, 306
 InvalidOptionGroupState, 875
 InvalidParameter, 621
 InvalidParameterCombinationFault, 686
 InvalidParameterException, 311, 316, 833
 InvalidParameterValue, 621
 InvalidParameterValueException, 192
 InvalidRequestContentException, 192
 InvalidRequestException, 246, 327
 InvalidRestore, 686, 875
 InvalidRestoreFault, 686
 InvalidRevisionException, 306
 InvalidRoleException, 306, 319
 InvalidS3BucketName, 686
 InvalidS3BucketNameException, 287
 InvalidS3KeyPrefix, 686
 InvalidS3KeyPrefixException, 319
 InvalidS3PrefixException, 287
 InvalidSequenceTokenException, 833
 InvalidSNSTopicARNException, 319
 InvalidSnsTopicNameException, 287
 InvalidSortByException, 306
 InvalidSortOrderException, 306
 InvalidSubnet, 686, 875
 InvalidSubscriptionState, 686
 InvalidTimeRangeException, 287, 306, 319
 InvalidTrailNameException, 287
 InvalidUriError, 219
 InvalidVPCNetworkState, 686, 876
 InvalidVPCNetworkStateFault, 686
 InventorySupplyList (boto.mws.response.ListInventorySupplyResult
 attribute), 626
 invoke_async() (boto.awslambda.layer1.AWSLambdaConnection
 method), 189
 IPPermissions (class in boto.ec2.securitygroup), 456
 IPPermissionsList (class in boto.ec2.securitygroup), 456
 IPRange (class in boto.rds.dbsecuritygroup), 652
 is_binary() (in module boto.dynamodb.types), 348
 is_cross_zone_load_balancing()
 (boto.ec2.elb.loadbalancer.LoadBalancer
 method), 504
 is_expired() (boto.sts.credentials.Credentials method),
 775
 is_num() (in module boto.dynamodb.types), 348
 is_str() (in module boto.dynamodb.types), 348
 is_stream() (boto.file.key.Key method), 517

- is_valid() (boto.mturk.question.QuestionForm method), 610
- isdir() (boto.manage.cmdshell.LocalClient method), 597
- isdir() (boto.manage.cmdshell.SSHClient method), 598
- Item (class in boto.dynamodb.item), 344
- Item (class in boto.dynamodb2.items), 353
- Item (class in boto.ecs.item), 470
- Item (class in boto.sdb.item), 741
- item_count (boto.dynamodb.table.Table attribute), 340
- item_lookup() (boto.ecs.ECSCConnection method), 470
- item_object_hook() (in module boto.dynamodb.types), 348
- item_search() (boto.ecs.ECSCConnection method), 470
- ItemAttributes (class in boto.mws.response), 626
- ItemCollectionSizeLimitExceededException, 402
- ItemData (boto.mws.response.ListInboundShipmentItemsResult attribute), 626
- ItemDimensions (boto.mws.response.AttributeSet attribute), 621
- ItemNotFound, 402
- ItemPrice (boto.mws.response.OrderItem attribute), 628
- items() (boto.dynamodb2.items.Item method), 354
- items() (boto.sqs.message.MHMessage method), 763
- items() (boto.utils.LazyLoadMetadata method), 222
- ItemSet (class in boto.ecs.item), 470
- ItemTax (boto.mws.response.OrderItem attribute), 628
- ItemThread (class in boto.sdb.connection), 733
- iter_call() (boto.mws.connection.MWSCConnection method), 617
- iter_response() (boto.mws.connection.MWSCConnection method), 617
- ## J
- jar() (boto.emr.step.JarStep method), 511
- jar() (boto.emr.step.Step method), 511
- jar() (boto.emr.step.StreamingStep method), 512
- JarStep (class in boto.emr.step), 511
- JavaApplet (class in boto.mturk.question), 608
- Job (class in boto.glacier.job), 536
- JobFlow (class in boto.emr.emrobject), 514
- JobFlowStepList (class in boto.emr.emrobject), 514
- json_request() (boto.swf.layer1.Layer1 method), 780
- JSONMessage (class in boto.sqs.jsonmessage), 761
- JSONResponseError, 219
- ## K
- key (boto.gs.key.Key attribute), 566
- key (boto.s3.key.Key attribute), 722
- Key (class in boto.file.key), 516
- Key (class in boto.gs.key), 559
- Key (class in boto.s3.key), 716
- Key (class in boto.sdb.db.key), 919
- KEY_REGULAR_FILE (boto.file.key.Key attribute), 516
- KEY_STREAM (boto.file.key.Key attribute), 516
- KEY_STREAM_READABLE (boto.file.key.Key attribute), 516
- KEY_STREAM_WRITABLE (boto.file.key.Key attribute), 516
- KeyPair (class in boto.ec2.keypair), 451
- keys() (boto.dynamodb2.items.Item method), 354
- keys() (boto.sqs.message.MHMessage method), 763
- KeysOnlyIndex (class in boto.dynamodb2.fields), 353
- KeyUnavailableException, 828
- KeyValue (class in boto.emr.emrobject), 514
- kind() (boto.sdb.db.key.Key method), 919
- KinesisConnection (class in boto.kinesis.layer1), 589
- KMSConnection (class in boto.kms.layer1), 822
- KMSInternalException, 828
- ## L
- LandedPrice (boto.mws.response.Price attribute), 628
- Languages (boto.mws.response.ItemAttributes attribute), 626
- LastDeliveryChannelDeleteFailedException, 320
- LaunchConfiguration (class in boto.beanstalk.response), 203
- LaunchConfiguration (class in boto.ec2.autoscale.launchconfig), 184
- LaunchSpecification (class in boto.ec2.launchspecification), 452
- Layer1 (class in boto.beanstalk.layer1), 192
- Layer1 (class in boto.cloudsearch.layer1), 250
- Layer1 (class in boto.dynamodb.layer1), 327
- Layer1 (class in boto.glacier.layer1), 521
- Layer1 (class in boto.swf.layer1), 777
- Layer1Decisions (class in boto.swf.layer1_decisions), 790
- Layer2 (class in boto.cloudsearch.layer2), 256
- Layer2 (class in boto.cloudsearch2.layer2), 272
- Layer2 (class in boto.dynamodb.layer2), 331
- Layer2 (class in boto.glacier.layer2), 533
- LayoutParameter (class in boto.mturk.layoutparam), 605
- LayoutParameters (class in boto.mturk.layoutparam), 605
- LazyLoadMetadata (class in boto.utils), 222
- LbAttributes (class in boto.ec2.elb.attributes), 506
- LBCookieStickinessPolicy (class in boto.ec2.elb.policies), 505
- LengthConstraint (class in boto.mturk.question), 608
- Lifecycle (class in boto.s3.lifecycle), 730
- LimitExceededException, 311, 316, 402, 495, 597, 828, 833, 846
- List (class in boto.mturk.question), 609
- list() (boto.gs.bucket.Bucket method), 550
- list() (boto.s3.bucket.Bucket method), 706
- list_activity_types() (boto.swf.layer1.Layer1 method), 781
- list_aliases() (boto.kms.layer1.KMSConnection method), 826

- list_all_fulfillment_orders()
(boto.mws.connection.MWSSConnection method), 617
- list_all_fulfillment_orders_by_next_token()
(boto.mws.connection.MWSSConnection method), 617
- list_all_parts() (boto.glacier.vault.Vault method), 534
- list_application_revisions()
(boto.coddeploy.layer1.CodeDeployConnection method), 300
- list_applications() (boto.coddeploy.layer1.CodeDeployConnection method), 300
- list_available_solution_stacks()
(boto.beanstalk.layer1.Layer1 method), 198
- list_available_zones() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
- list_bootstrap_actions() (boto.emr.connection.EmrConnection method), 508
- list_carts() (boto.mws.connection.MWSSConnection method), 617
- list_carts_by_next_token()
(boto.mws.connection.MWSSConnection method), 618
- list_closed_workflow_executions()
(boto.swf.layer1.Layer1 method), 781
- list_clusters() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 466
- list_clusters() (boto.emr.connection.EmrConnection method), 508
- list_container_instances()
(boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 466
- list_customers() (boto.mws.connection.MWSSConnection method), 618
- list_customers_by_next_token()
(boto.mws.connection.MWSSConnection method), 618
- list_datasets() (boto.cognito.sync.layer1.CognitoSyncConnection method), 313
- list_deployment_configs()
(boto.coddeploy.layer1.CodeDeployConnection method), 300
- list_deployment_groups()
(boto.coddeploy.layer1.CodeDeployConnection method), 300
- list_deployment_instances()
(boto.coddeploy.layer1.CodeDeployConnection method), 301
- list_deployments() (boto.coddeploy.layer1.CodeDeployConnection method), 301
- list_domain_names() (boto.cloudsearch2.layer1.CloudSearchConnection method), 271
- list_domains() (boto.cloudsearch.layer2.Layer2 method), 256
- list_domains() (boto.cloudsearch2.layer2.Layer2 method), 272
- list_domains() (boto.route53.domains.layer1.Route53DomainsConnection method), 911
- list_domains() (boto.swf.layer1.Layer1 method), 782
- list_endpoints_by_platform_application()
(boto.sns.SNSConnection method), 751
- list_entities_for_policy() (boto.iam.connection.IAMConnection method), 582
- list_event_sources() (boto.awslambda.layer1.AWSLambdaConnection method), 189
- list_functions() (boto.awslambda.layer1.AWSLambdaConnection method), 190
- list_grants() (boto.gs.bucket.Bucket method), 551
- list_grants() (boto.kms.layer1.KMSConnection method), 826
- list_grants() (boto.s3.bucket.Bucket method), 707
- list_hapgs() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
- list_hsms() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
- list_identities() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 309
- list_identities() (boto.ses.connection.SESConnection method), 746
- list_identity_pools() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 309
- list_identity_pools() (boto.cognito.sync.layer1.CognitoSyncConnection method), 313
- list_identity_pools() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 309
- list_inbound_shipment_items()
(boto.mws.connection.MWSSConnection method), 618
- list_inbound_shipment_items_by_next_token()
(boto.mws.connection.MWSSConnection method), 618
- list_inbound_shipments()
(boto.mws.connection.MWSSConnection method), 618
- list_inbound_shipments_by_next_token()
(boto.mws.connection.MWSSConnection method), 618
- list_instance_groups() (boto.emr.connection.EmrConnection method), 508
- list_instance_profiles() (boto.iam.connection.IAMConnection method), 583
- list_instance_profiles_for_role()
(boto.iam.connection.IAMConnection method), 583
- list_instances() (boto.emr.connection.EmrConnection method), 508
- list_inventory_supply() (boto.mws.connection.MWSSConnection method), 618
- list_inventory_supply_by_next_token()

- (boto.mws.connection.MWSConnection method), 618
- list_jobs() (boto.glacier.layer1.Layer1 method), 528
- list_jobs() (boto.glacier.vault.Vault method), 534
- list_jobs_by_pipeline() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 490
- list_jobs_by_status() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 490
- list_key_policies() (boto.kms.layer1.KMSConnection method), 826
- list_keys() (boto.kms.layer1.KMSConnection method), 826
- list_luna_clients() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
- list_marketplace_participations() (boto.mws.connection.MWSConnection method), 618
- list_marketplace_participations_by_next_token() (boto.mws.connection.MWSConnection method), 619
- list_matching_products() (boto.mws.connection.MWSConnection method), 619
- list_metrics() (boto.ec2.cloudwatch.CloudWatchConnection method), 290
- list_multipart_uploads() (boto.glacier.layer1.Layer1 method), 529
- list_multipart_uploads() (boto.gs.bucket.Bucket method), 551
- list_multipart_uploads() (boto.s3.bucket.Bucket method), 707
- list_open_workflow_executions() (boto.swf.layer1.Layer1 method), 783
- list_operations() (boto.route53.domains.layer1.Route53DomainsConnection method), 912
- list_order_items() (boto.mws.connection.MWSConnection method), 619
- list_order_items_by_next_token() (boto.mws.connection.MWSConnection method), 619
- list_orders() (boto.mws.connection.MWSConnection method), 619
- list_orders_by_next_token() (boto.mws.connection.MWSConnection method), 619
- list_parts() (boto.glacier.layer1.Layer1 method), 529
- list_pipelines() (boto.datapipeline.layer1.DataPipelineConnection method), 324
- list_pipelines() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 490
- list_platform_applications() (boto.sns.SNSConnection method), 752
- list_policies() (boto.iam.connection.IAMConnection method), 583
- list_policy_versions() (boto.iam.connection.IAMConnection method), 583
- list_presets() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 490
- list_records() (boto.cognito.sync.layer1.CognitoSyncConnection method), 314
- list_registered_destinations() (boto.mws.connection.MWSConnection method), 620
- list_resources() (boto.cloudformation.stack.Stack method), 233
- list_role_policies() (boto.iam.connection.IAMConnection method), 583
- list_roles() (boto.iam.connection.IAMConnection method), 584
- list_saml_providers() (boto.iam.connection.IAMConnection method), 584
- list_server_certs() (boto.iam.connection.IAMConnection method), 584
- list_stack_resources() (boto.cloudformation.connection.CloudFormationConnection method), 229
- list_stacks() (boto.cloudformation.connection.CloudFormationConnection method), 230
- list_steps() (boto.emr.connection.EmrConnection method), 508
- list_streams() (boto.kinesis.layer1.KinesisConnection method), 592
- list_subscriptions() (boto.mws.connection.MWSConnection method), 620
- list_tables() (boto.dynamodb.layer1.Layer1 method), 329
- list_tables() (boto.dynamodb.layer2.Layer2 method), 334
- list_tables() (boto.dynamodb2.layer1.DynamoDBConnection method), 381
- list_tags_for_resource() (boto.rds2.layer1.RDSConnection method), 897
- list_tags_for_stream() (boto.kinesis.layer1.KinesisConnection method), 593
- list_task_definitions() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 466
- list_tasks() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 467
- list_vaults() (boto.glacier.layer1.Layer1 method), 530
- list_vaults() (boto.glacier.layer2.Layer2 method), 533
- list_verified_email_addresses() (boto.ses.connection.SESConnection method), 746
- list_versions() (boto.gs.bucket.Bucket method), 552
- list_versions() (boto.s3.bucket.Bucket method), 707
- list_workflow_types() (boto.swf.layer1.Layer1 method),

- 783
- ListAllFulfillmentOrdersResult (class in boto.mws.response), 626
- ListAvailableSolutionStacksResponse (class in boto.beanstalk.response), 203
- ListCartsResult (class in boto.mws.response), 626
- ListCustomersResult (class in boto.mws.response), 626
- listdir() (boto.manage.cmdshell.LocalClient method), 598
- listdir() (boto.manage.cmdshell.SSHClient method), 599
- ListElement (class in boto.ec2.elb.listelement), 502
- Listener (class in boto.beanstalk.response), 203
- Listener (class in boto.ec2.elb.listener), 502
- ListInboundShipmentItemsResult (class in boto.mws.response), 626
- ListInboundShipmentsResult (class in boto.mws.response), 626
- ListingPrice (boto.mws.response.Price attribute), 628
- ListingQualityRecommendations (boto.mws.response.ListRecommendationsResult attribute), 627
- ListInventorySupplyResult (class in boto.mws.response), 626
- ListMarketplaceParticipationsResult (class in boto.mws.response), 626
- ListMarketplaces (boto.mws.response.ListMarketplaceParticipationsResult attribute), 626
- ListMatchingProductsResult (class in boto.mws.response), 626
- ListOrderItemsResult (class in boto.mws.response), 627
- ListOrdersResult (class in boto.mws.response), 627
- ListParticipations (boto.mws.response.ListMarketplaceParticipationsResult attribute), 626
- ListPrice (boto.mws.response.AttributeSet attribute), 621
- ListProperty (class in boto.sdb.db.property), 923
- ListRecommendationsResult (class in boto.mws.response), 627
- ListRegisteredDestinationsResult (class in boto.mws.response), 627
- ListSubscriptionsResult (class in boto.mws.response), 627
- load() (boto.dynamodb2.items.Item method), 354
- load() (boto.sdb.item.Item method), 742
- load() (boto.sqs.queue.Queue method), 765
- load() (boto.sts.credentials.Credentials class method), 776
- load_boto() (boto.pyami.bootstrap.Bootstrap method), 630
- load_credential_file() (boto.pyami.config.Config method), 631
- load_from_file() (boto.sqs.queue.Queue method), 765
- load_from_filename() (boto.sqs.queue.Queue method), 765
- load_from_path() (boto.pyami.config.Config method), 631
- load_from_s3() (boto.sqs.queue.Queue method), 765
- load_from_sdb() (boto.pyami.config.Config method), 631
- load_object() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
- load_packages() (boto.pyami.bootstrap.Bootstrap method), 630
- LoadBalancer (class in boto.beanstalk.response), 203
- LoadBalancer (class in boto.ec2.elb.loadbalancer), 502
- LoadBalancerDescription (class in boto.beanstalk.response), 203
- LoadBalancerZones (class in boto.ec2.elb.loadbalancer), 505
- LocalClient (class in boto.manage.cmdshell), 597
- LocaleRequirement (class in boto.mturk.qualification), 606
- Location (class in boto.gs.connection), 559
- Location (class in boto.s3.connection), 710
- log_message() (boto.services.result.ResultProcessor method), 743
- LogFileName (boto.services.result.ResultProcessor attribute), 743
- LoggingGroup (boto.s3.bucket.Bucket attribute), 698
- LoggingInfo (class in boto.cloudfront.logging), 243
- LongProperty (class in boto.sdb.db.property), 923
- lookup() (boto.cloudsearch.layer2.Layer2 method), 256
- lookup() (boto.cloudsearch2.layer2.Layer2 method), 272
- lookup() (boto.dynamodb.layer2.Layer2 method), 334
- lookup() (boto.dynamodb.table.Table method), 340
- lookup() (boto.dynamodb2.table.Table method), 364
- lookup() (boto.gs.bucket.Bucket method), 552
- lookup() (boto.gs.connection.GSConnection method), 558
- lookup() (boto.s3.bucket.Bucket method), 708
- lookup() (boto.s3.connection.S3Connection method), 713
- lookup() (boto.sdb.connection.SDBConnection method), 737
- lookup() (boto.sqs.connection.SQSConnection method), 758
- lookup_developer_identity() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 309
- lookup_events() (boto.cloudtrail.layer1.CloudTrailConnection method), 285
- LossyFloatDynamizer (class in boto.dynamodb.types), 348
- LowestOfferListing (class in boto.mws.response), 627
- LowestOfferListings (boto.mws.response.Product attribute), 628
- LRUCache (class in boto.utils), 221

M

- MachineLearningConnection (class in boto.machinelearning.layer1), 833
- main() (boto.pyami.bootstrap.Bootstrap method), 630

- main() (boto.pyami.copybot.CopyBot method), 631
- main() (boto.pyami.installers.ubuntu.apache.Apache method), 632
- main() (boto.pyami.installers.ubuntu.mysql.MySQL method), 633
- main() (boto.pyami.installers.ubuntu.trac.Trac method), 633
- main() (boto.pyami.scriptbase.ScriptBase method), 633
- main() (boto.pyami.startup.Startup method), 633
- main() (boto.services.bs.BS method), 743
- main() (boto.services.service.Service method), 744
- main() (in module boto.pyami.launch_ami), 633
- main_class() (boto.emr.step.JarStep method), 511
- main_class() (boto.emr.step.Step method), 512
- main_class() (boto.emr.step.StreamingStep method), 512
- make_public() (boto.gs.bucket.Bucket method), 552
- make_public() (boto.gs.key.Key method), 566
- make_public() (boto.s3.bucket.Bucket method), 708
- make_public() (boto.s3.key.Key method), 722
- make_request() (boto.awslambda.layer1.AWSLambdaConnection method), 190
- make_request() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
- make_request() (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection method), 278
- make_request() (boto.cloudtrail.layer1.CloudTrailConnection method), 285
- make_request() (boto.codedeploy.layer1.CodeDeployConnection method), 301
- make_request() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 310
- make_request() (boto.cognito.sync.layer1.CognitoSyncConnection method), 314
- make_request() (boto.configservice.layer1.ConfigServiceConnection method), 318
- make_request() (boto.connection.AWSAuthConnection method), 215
- make_request() (boto.connection.AWSQueryConnection method), 216
- make_request() (boto.datapipeline.layer1.DataPipelineConnection method), 324
- make_request() (boto.dynamodb.layer1.Layer1 method), 329
- make_request() (boto.dynamodb2.layer1.DynamoDBConnection method), 381
- make_request() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491
- make_request() (boto.glacier.layer1.Layer1 method), 530
- make_request() (boto.gs.connection.GSConnection method), 558
- make_request() (boto.kinesis.layer1.KinesisConnection method), 593
- make_request() (boto.kms.layer1.KMSCConnection method), 827
- make_request() (boto.logs.layer1.CloudWatchLogsConnection method), 831
- make_request() (boto.machinelearning.layer1.MachineLearningConnection method), 844
- make_request() (boto.opsworks.layer1.OpsWorksConnection method), 864
- make_request() (boto.route53.connection.Route53Connection method), 691
- make_request() (boto.route53.domains.layer1.Route53DomainsConnection method), 912
- make_request() (boto.s3.connection.S3Connection method), 713
- make_request() (boto.support.layer1.SupportConnection method), 931
- make_request() (boto.swf.layer1.Layer1 method), 784
- make_value_from_datastore() (boto.sdb.db.property.PasswordProperty method), 924
- make_value_from_datastore() (boto.sdb.db.property.Property method), 924
- MalformedPolicyDocumentException, 828
- manage_report_schedule() (boto.mws.connection.MWSConnection method), 620
- ManageReportScheduleResult (class in boto.mws.response), 627
- MapProperty (class in boto.sdb.db.property), 923
- mark_clean() (boto.dynamodb2.items.Item method), 354
- MarkDirty() (boto.dynamodb2.items.Item method), 355
- MaterialType (boto.mws.response.VariationRelationship attribute), 630
- max_batch_get (boto.dynamodb2.table.Table attribute), 364
- MAX_SELECTION_COUNT_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610
- MaximumNumberOfTrailsExceededException, 287
- MaxNumberOfConfigurationRecordersExceededException, 320
- MaxNumberOfDeliveryChannelsExceededException, 320
- md5 (boto.gs.key.Key attribute), 566
- md5 (boto.s3.key.Key attribute), 722
- MemberList (class in boto.mws.response), 627
- merge() (boto.rds.parametergroup.ParameterGroup method), 653
- merge_developer_identities() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 310
- merge_headers_by_name() (in module boto.utils), 223
- merge_meta() (in module boto.utils), 224
- merge_shards() (boto.kinesis.layer1.KinesisConnection method), 593

Message (class in boto.sqs.message), 763
 Messages (boto.mws.response.GetServiceStatusResult attribute), 626
 MessageSet (class in boto.ec2.zone), 463
 method_for() (boto.mws.connection.MWSConnection method), 620
 Metric (class in boto.ec2.cloudwatch.metric), 292
 MetricAlarm (class in boto.ec2.cloudwatch.alarm), 294
 MetricAlarms (class in boto.ec2.cloudwatch.alarm), 295
 MetricCollectionTypes (class in boto.ec2.autoscale.policy), 186
 MetricCollectionTypes.BaseType (class in boto.ec2.autoscale.policy), 186
 MetricCollectionTypes.Granularity (class in boto.ec2.autoscale.policy), 186
 MetricCollectionTypes.Metric (class in boto.ec2.autoscale.policy), 186
 MFADeleteRE (boto.s3.bucket.Bucket attribute), 698
 MHMessage (class in boto.sqs.message), 762
 MIN_ETAG_LEN (boto.s3.resumable_download_handler.ResumableDownloadHandler attribute), 729
 MIN_SELECTION_COUNT_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610
 mkdir() (boto.pyami.scriptbase.ScriptBase method), 633
 mklst() (in module boto.utils), 224
 modifiable() (boto.rds.parametergroup.ParameterGroup method), 654
 ModificationResult (class in boto.ec2.reservedinstance), 454
 modify() (boto.rds.dbinstance.DBInstance method), 649
 modify_attribute() (boto.ec2.instance.Instance method), 447
 modify_cache_cluster() (boto.elasticache.layer1.ElastiCacheConnection method), 481
 modify_cache_parameter_group() (boto.elasticache.layer1.ElastiCacheConnection method), 482
 modify_cache_subnet_group() (boto.elasticache.layer1.ElastiCacheConnection method), 482
 modify_cluster() (boto.redshift.layer1.RedshiftConnection method), 673
 modify_cluster_parameter_group() (boto.redshift.layer1.RedshiftConnection method), 675
 modify_cluster_subnet_group() (boto.redshift.layer1.RedshiftConnection method), 676
 modify_db_instance() (boto.rds2.layer1.RDSConnection method), 898
 modify_db_parameter_group() (boto.rds2.layer1.RDSConnection method), 902
 modify_db_subnet_group() (boto.rds.RDSConnection method), 643
 modify_db_subnet_group() (boto.rds2.layer1.RDSConnection method), 902
 modify_dbinstance() (boto.rds.RDSConnection method), 643
 modify_event_subscription() (boto.rds2.layer1.RDSConnection method), 902
 modify_event_subscription() (boto.redshift.layer1.RedshiftConnection method), 676
 modify_hapg() (boto.cloudhsm.layer1.CloudHSMConnection method), 245
 modify_hsm() (boto.cloudhsm.layer1.CloudHSMConnection method), 246
 modify_image_attribute() (boto.ec2.connection.EC2Connection method), 427
 modify_instance_attribute() (boto.ec2.connection.EC2Connection method), 427
 modify_instance_groups() (boto.emr.connection.EmrConnection method), 509
 modify_lb_attribute() (boto.ec2.elb.ELBConnection method), 500
 modify_luna_client() (boto.cloudhsm.layer1.CloudHSMConnection method), 246
 modify_network_interface_attribute() (boto.ec2.connection.EC2Connection method), 428
 modify_notification_group() (boto.rds2.layer1.RDSConnection method), 903
 modify_parameter_group() (boto.rds.RDSConnection method), 645
 modify_replication_group() (boto.elasticache.layer1.ElastiCacheConnection method), 483
 modify_reserved_instances() (boto.ec2.connection.EC2Connection method), 428
 modify_snapshot_attribute() (boto.ec2.connection.EC2Connection method), 428
 modify_snapshot_copy_retention_period() (boto.redshift.layer1.RedshiftConnection method), 677
 modify_volume_attribute() (boto.ec2.connection.EC2Connection method), 429
 modify_vpc_attribute() (boto.ec2.connection.EC2Connection method), 429

- [modify_vpc_attribute\(\)](#) (boto.vpc.VPCConnection method), 815
[ModifyInstanceGroupsResponse](#) (class in boto.emr.emrobject), 514
[ModifyReservedInstancesResult](#) (class in boto.ec2.reservedinstance), 454
[monitor\(\)](#) (boto.ec2.instance.Instance method), 447
[monitor_instance\(\)](#) (boto.ec2.connection.EC2Connection method), 429
[monitor_instances\(\)](#) (boto.ec2.connection.EC2Connection method), 429
[MSG](#) (boto.exception.InvalidInstanceMetadataError attribute), 218
[MTurkConnection](#) (class in boto.mturk.connection), 601
[MTurkRequestError](#), 604
[MultiDeleteResult](#) (class in boto.s3.multidelete), 729
[multipart_upload_lister\(\)](#) (in module boto.s3.bucketlistresultset), 710
[MultiPartUpload](#) (class in boto.s3.multipart), 727
[MultiPartUploadListResultSet](#) (class in boto.s3.bucketlistresultset), 710
[MWSConnection](#) (class in boto.mws.connection), 611
[MWSConnection.ResponseErrorFactory](#) (class in boto.mws.connection), 611
[MWSConnection.ResponseFactory](#) (class in boto.mws.connection), 611
[MySQL](#) (class in boto.pyami.installers.ubuntu.mysql), 632
- ## N
- [name](#) (boto.cloudsearch.domain.Domain attribute), 249
[name](#) (boto.cloudsearch2.domain.Domain attribute), 265
[name](#) (boto.dynamodb.table.Table attribute), 340
[name](#) (boto.sdb.db.property.Property attribute), 924
[name](#) (boto.sqs.queue.Queue attribute), 765
[name\(\)](#) (boto.sdb.db.key.Key method), 919
[NameSpace](#) (boto.s3.acl.Grant attribute), 698
[needs_save\(\)](#) (boto.dynamodb2.items.Item method), 355
[NetworkInterface](#) (class in boto.ec2.networkinterface), 452
[NetworkInterfaceCollection](#) (class in boto.ec2.networkinterface), 453
[NetworkInterfaceSpecification](#) (class in boto.ec2.networkinterface), 453
[new_batch_list\(\)](#) (boto.dynamodb.layer2.Layer2 method), 334
[new_batch_write_list\(\)](#) (boto.dynamodb.layer2.Layer2 method), 334
[new_http_connection\(\)](#) (boto.connection.AWSAuthConnection method), 215
[new_http_connection\(\)](#) (boto.gs.connection.GSConnection method), 558
[new_item\(\)](#) (boto.dynamodb.table.Table method), 340
[new_item\(\)](#) (boto.dynamodb2.table.Table method), 364
[new_item\(\)](#) (boto.sdb.domain.Domain method), 740
[new_key\(\)](#) (boto.file.bucket.Bucket method), 516
[new_key\(\)](#) (boto.gs.bucket.Bucket method), 552
[new_key\(\)](#) (boto.s3.bucket.Bucket method), 708
[new_message\(\)](#) (boto.sqs.queue.Queue method), 765
[new_statement\(\)](#) (boto.cloudsearch.optionstatus.ServicePoliciesStatus method), 258
[new_statement\(\)](#) (boto.cloudsearch2.optionstatus.ServicePoliciesStatus method), 273
[NEWVALUE](#) (class in boto.dynamodb2.items), 356
[next\(\)](#) (boto.dynamodb2.results.ResultSet method), 356
[next\(\)](#) (boto.ecs.item.ItemSet method), 470
[next\(\)](#) (boto.gs.key.Key method), 566
[next\(\)](#) (boto.s3.key.Key method), 722
[next\(\)](#) (boto.sdb.db.blob.Blob method), 918
[next\(\)](#) (boto.sdb.db.query.Query method), 926
[next\(\)](#) (boto.sdb.queryresultset.SelectResultSet method), 742
[next_page\(\)](#) (boto.cloudsearch.search.SearchResults method), 260
[next_page\(\)](#) (boto.cloudsearch2.search.SearchResults method), 276
[next_response\(\)](#) (boto.dynamodb.layer2.TableGenerator method), 337
[next_token](#) (boto.sdb.db.query.Query attribute), 926
[NoAuthHandlerFound](#), 219
[NoAvailableConfigurationRecorderException](#), 320
[NoAvailableDeliveryChannelException](#), 320
[NoHostProvided](#) (class in boto.s3.connection), 711
[NonBooleanDynamizer](#) (class in boto.dynamodb.types), 348
[NoRunningConfigurationRecorderException](#), 320
[NoSuchBucketException](#), 320
[NoSuchConfigurationRecorderException](#), 320
[NoSuchDeliveryChannelException](#), 320
[NotAuthorizedException](#), 311, 316
[NotFoundException](#), 828
[NOTIFICATION_VERSION](#)
 (boto.mturk.notification.NotificationMessage attribute), 605
[NOTIFICATION_WSDL](#)
 (boto.mturk.notification.NotificationMessage attribute), 605
[NotificationARN](#) (class in boto.cloudformation.stack), 232
[NotificationEmailList](#) (boto.mws.response.FulfillmentOrder attribute), 624
[NotificationMessage](#) (class in boto.mturk.notification), 605
[notify\(\)](#) (boto.pyami.scriptbase.ScriptBase method), 633
[notify\(\)](#) (in module boto.utils), 224
[notify_workers\(\)](#) (boto.mturk.connection.MTurkConnection method), 603
[now\(\)](#) (boto.sdb.db.property.DateProperty method), 922

- now() (boto.sdb.db.property.DateTimeProperty method), 922
 - NullHandler (class in boto), 204
 - num_searchable_docs (boto.cloudsearch.domain.Domain attribute), 250
 - NumberHitsApprovedRequirement (class in boto.mturk.qualification), 606
 - NumberOfLinesSuggestion (class in boto.mturk.question), 609
 - NumberOfNodesPerClusterLimitExceeded, 686
 - NumberOfNodesPerClusterLimitExceededFault, 686
 - NumberOfNodesQuotaExceeded, 686
 - NumberOfNodesQuotaExceededFault, 686
 - NumberOfOfferListings (boto.mws.response.CompetitivePricing attribute), 622
 - NumberRetries (boto.dynamodb.layer1.Layer1 attribute), 328
 - NumberRetries (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370
 - NumericConstraint (class in boto.mturk.question), 609
- O**
- Object (class in boto.cloudfront.object), 242
 - Offer (class in boto.mws.response), 627
 - Offers (boto.mws.response.Product attribute), 628
 - OK (boto.ec2.cloudwatch.alarm.MetricAlarm attribute), 295
 - open() (boto.gs.key.Key method), 566
 - open() (boto.manage.cmdshell.SSHClient method), 599
 - open() (boto.s3.key.Key method), 722
 - open_read() (boto.gs.key.Key method), 566
 - open_read() (boto.s3.key.Key method), 722
 - open_sftp() (boto.manage.cmdshell.SSHClient method), 599
 - open_write() (boto.gs.key.Key method), 567
 - open_write() (boto.s3.key.Key method), 722
 - OperatingSystem (boto.mws.response.VariationRelationship attribute), 630
 - OPERATION_NAME (boto.mturk.notification.NotificationMessage attribute), 606
 - OperationAbortedException, 833
 - OperationLimitExceeded, 918
 - OpsWorksConnection (class in boto.opsworks.layer1), 847
 - OptionGroupAlreadyExists, 876
 - OptionGroupNotFound, 876
 - OptionGroupQuotaExceeded, 876
 - OptionRestrictionRegex (class in boto.beanstalk.response), 203
 - OptionStatus (class in boto.cloudsearch.optionstatus), 257
 - OptionStatus (class in boto.cloudsearch2.optionstatus), 272
 - Order (class in boto.mws.response), 627
 - order() (boto.sdb.db.query.Query method), 926
 - OrderedContent (class in boto.mturk.question), 609
 - OrderItem (class in boto.mws.response), 627
 - OrderItems (boto.mws.response.ListOrderItemsResult attribute), 627
 - OrderReferenceDetails (boto.mws.response.SetOrderReferenceDetailsResult attribute), 629
 - OrderReferenceDetails (class in boto.mws.response), 628
 - OrderReferenceStatus (boto.mws.response.OrderReferenceDetails attribute), 628
 - Orders (boto.mws.response.ListOrdersResult attribute), 627
 - OrderTotal (boto.mws.response.Order attribute), 627
 - OrderTotal (boto.mws.response.OrderReferenceDetails attribute), 628
 - OrdinaryCallingFormat (class in boto.s3.connection), 711
 - OriginAccessIdentity (class in boto.cloudfront.identity), 240
 - OriginAccessIdentityConfig (class in boto.cloudfront.identity), 240
 - OriginAccessIdentitySummary (class in boto.cloudfront.identity), 240
 - OTHER_SELECTION_ELEMENT_NAME (boto.mturk.question.SelectionAnswer attribute), 610
 - OtherPolicy (class in boto.ec2.elb.policies), 505
 - output (boto.utils.ShellCommand attribute), 222
 - Output (class in boto.cloudformation.stack), 232
 - Overview (class in boto.mturk.question), 609
- P**
- PackageDimensions (boto.mws.response.AttributeSet attribute), 621
 - Parameter (class in boto.cloudformation.stack), 232
 - Parameter (class in boto.rds.parametergroup), 653
 - parameter_group (boto.rds.dbinstance.DBInstance attribute), 650
 - parameter_template (boto.mturk.question.Application attribute), 607
 - ParameterGroup (class in boto.rds.parametergroup), 654
 - parent() (boto.sdb.db.key.Key method), 919
 - parse_host() (in module boto.utils), 224
 - parse_ts() (in module boto.utils), 224
 - parseString() (boto.handler.XmlHandlerWrapper method), 220
 - Part (class in boto.s3.multipart), 728
 - part_list() (in module boto.s3.multipart), 728
 - partial_save() (boto.dynamodb2.items.Item method), 355
 - Password (class in boto.utils), 222
 - PasswordProperty (class in boto.sdb.db.property), 923
 - pay() (boto.fps.connection.FPSConnection method), 520
 - PaymentExecutionDetail (boto.mws.response.Order attribute), 627

- PendingModifiedValues (class in boto.rds.dbinstance), 651
- PercentAssignmentsAbandonedRequirement (class in boto.mturk.qualification), 606
- PercentAssignmentsApprovedRequirement (class in boto.mturk.qualification), 606
- PercentAssignmentsRejectedRequirement (class in boto.mturk.qualification), 607
- PercentAssignmentsReturnedRequirement (class in boto.mturk.qualification), 607
- PercentAssignmentsSubmittedRequirement (class in boto.mturk.qualification), 607
- PigBase (class in boto.emr.step), 511
- PigStep (class in boto.emr.step), 511
- PipelineDeletedException, 327
- PipelineNotFoundException, 327
- placement (boto.ec2.instance.Instance attribute), 447
- placement_group (boto.ec2.instance.Instance attribute), 447
- placement_tenancy (boto.ec2.instance.Instance attribute), 447
- PlacementGroup (class in boto.ec2.placementgroup), 454
- PleaseRetryException, 219
- PointInTimeRestoreNotEnabled, 876
- Policies (class in boto.ec2.elb.policies), 505
- Policy (class in boto.s3.acl), 698
- poll() (boto.swf.layer2.ActivityWorker method), 793
- poll() (boto.swf.layer2.Decider method), 794
- poll_for_activity_task() (boto.swf.layer1.Layer1 method), 784
- poll_for_decision_task() (boto.swf.layer1.Layer1 method), 784
- poll_for_task() (boto.datapipeline.layer1.DataPipelineConnection method), 324
- POSTHCXMLBody (boto.route53.connection.Route53Connection attribute), 688
- POSTXMLBody (boto.route53.healthcheck.HealthCheck attribute), 692
- predict() (boto.machinelearning.layer1.MachineLearningConnection method), 844
- PredictorNotMountedException, 846
- Prefix (class in boto.s3.prefix), 727
- prefix_proxy_to_path() (boto.connection.AWSAuthConnection method), 215
- prefix_proxy_to_path() (boto.gs.connection.GSConnection method), 558
- prepare_full() (boto.dynamodb2.items.Item method), 355
- prepare_partial() (boto.dynamodb2.items.Item method), 355
- previous_state (boto.ec2.instance.Instance attribute), 447
- previous_state_code (boto.ec2.instance.Instance attribute), 447
- Price (boto.mws.response.CompetitivePrice attribute), 622
- Price (boto.mws.response.LowestOfferListing attribute), 627
- Price (class in boto.mturk.price), 606
- Price (class in boto.mws.response), 628
- PriceSchedule (class in boto.ec2.reservedinstance), 454
- PricingDetail (class in boto.ec2.reservedinstance), 454
- PrimaryContactInfo (boto.mws.response.Customer attribute), 623
- print_command_help() (boto.services.bs.BS method), 743
- print_usage() (boto.sdb.connection.SDBConnection method), 737
- PrivateIPAddress (class in boto.ec2.networkinterface), 453
- process_file() (boto.services.service.Service method), 744
- process_file() (boto.services.sonofmmm.SonOfMMM method), 744
- process_record() (boto.services.result.ResultProcessor method), 743
- processing (boto.cloudsearch.domain.Domain attribute), 250
- processing (boto.cloudsearch2.domain.Domain attribute), 265
- ProcessingTime (boto.services.service.Service attribute), 744
- ProcessType (class in boto.ec2.autoscale.group), 183
- Product (boto.mws.response.ProductsBulkOperationResult attribute), 628
- Product (class in boto.mws.response), 628
- ProductCategory (class in boto.mws.response), 628
- ProductCodes (class in boto.ec2.image), 444
- Products (boto.mws.response.ListMatchingProductsResult attribute), 626
- ProductsBulkOperationResponse (class in boto.mws.response), 628
- ProductsBulkOperationResult (class in boto.mws.response), 628
- profile_name (boto.connection.AWSAuthConnection attribute), 215
- profile_name (boto.gs.connection.GSConnection attribute), 558
- projection_type (boto.dynamodb2.fields.AllIndex attribute), 349
- projection_type (boto.dynamodb2.fields.GlobalAllIndex attribute), 350
- projection_type (boto.dynamodb2.fields.GlobalIncludeIndex attribute), 351
- projection_type (boto.dynamodb2.fields.GlobalKeysOnlyIndex attribute), 352
- projection_type (boto.dynamodb2.fields.IncludeIndex attribute), 352
- projection_type (boto.dynamodb2.fields.KeysOnlyIndex attribute), 353

- promote_read_replica() (boto.rds.RDSCONNECTION method), 645
 - promote_read_replica() (boto.rds2.layer1.RDSCONNECTION method), 903
 - PromotionDiscount (boto.mws.response.OrderItem attribute), 628
 - PromotionIds (boto.mws.response.OrderItem attribute), 628
 - Property (class in boto.sdb.db.property), 924
 - ProtocolIndependentOrdinaryCallingFormat (class in boto.s3.connection), 711
 - provider (boto.gs.key.Key attribute), 567
 - provider (boto.s3.key.Key attribute), 722
 - provider (boto.s3.prefix.Prefix attribute), 727
 - ProvisionedIopsNotAvailableInAZ, 876
 - ProvisionedThroughputExceededException, 402, 597
 - proxy_ssl() (boto.connection.AWSAuthConnection method), 215
 - proxy_ssl() (boto.gs.connection.GSConnection method), 558
 - publish() (boto.sns.SNSConnection method), 752
 - purchase() (boto.ec2.reservedinstance.ReservedInstancesOffering method), 456
 - purchase_reserved_cache_nodes_offering() (boto.elasticache.layer1.ElastiCacheConnection method), 484
 - purchase_reserved_db_instances_offering() (boto.rds2.layer1.RDSCONNECTION method), 904
 - purchase_reserved_instance_offering() (boto.ec2.connection.EC2Connection method), 429
 - purchase_reserved_node_offering() (boto.redshift.layer1.RedshiftConnection method), 677
 - purge() (boto.sqs.queue.Queue method), 766
 - purge_queue() (boto.sqs.connection.SQSConnection method), 758
 - put() (boto.connection.HostConnectionPool method), 217
 - put() (boto.dynamodb.item.Item method), 345
 - put_attribute() (boto.dynamodb.item.Item method), 345
 - put_attributes() (boto.sdb.connection.SDBCConnection method), 737
 - put_attributes() (boto.sdb.domain.Domain method), 740
 - put_configuration_recorder() (boto.configservice.layer1.ConfigServiceConnection method), 318
 - put_delivery_channel() (boto.configservice.layer1.ConfigServiceConnection method), 318
 - put_file() (boto.manage.cmdshell.LocalClient method), 598
 - put_file() (boto.manage.cmdshell.SSHClient method), 599
 - put_file() (boto.services.service.Service method), 744
 - put_group_policy() (boto.iam.connection.IAMConnection method), 584
 - put_http_connection() (boto.connection.AWSAuthConnection method), 215
 - put_http_connection() (boto.connection.ConnectionPool method), 216
 - put_http_connection() (boto.gs.connection.GSConnection method), 558
 - put_item() (boto.dynamodb.layer1.Layer1 method), 329
 - put_item() (boto.dynamodb.layer2.Layer2 method), 334
 - put_item() (boto.dynamodb2.layer1.DynamoDBConnection method), 381
 - put_item() (boto.dynamodb2.table.BatchTable method), 357
 - put_item() (boto.dynamodb2.table.Table method), 364
 - put_key_policy() (boto.kms.layer1.KMSCONNECTION method), 827
 - put_log_events() (boto.logs.layer1.CloudWatchLogsConnection method), 831
 - put_metric_alarm() (boto.ec2.cloudwatch.CloudWatchConnection method), 291
 - put_metric_data() (boto.ec2.cloudwatch.CloudWatchConnection method), 291
 - put_metric_filter() (boto.logs.layer1.CloudWatchLogsConnection method), 832
 - put_notification_configuration() (boto.ec2.autoscale.AutoScaleConnection method), 180
 - put_notification_configuration() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 - put_pipeline_definition() (boto.datapipeline.layer1.DataPipelineConnection method), 324
 - put_record() (boto.kinesis.layer1.KinesisConnection method), 593
 - put_records() (boto.kinesis.layer1.KinesisConnection method), 594
 - put_retention_policy() (boto.logs.layer1.CloudWatchLogsConnection method), 832
 - put_role_policy() (boto.iam.connection.IAMConnection method), 584
 - put_user_policy() (boto.iam.connection.IAMConnection method), 585
 - pythonize_name() (in module boto.utils), 224
- Q**
- Qualification (class in boto.mturk.connection), 604
 - QualificationRequest (class in boto.mturk.connection), 605
 - Qualifications (class in boto.mturk.qualification), 607
 - QualificationType (class in boto.mturk.connection), 605
 - Qualifiers (boto.mws.response.LowestOfferListing attribute), 627

- Query (class in boto.cloudsearch.search), 258
 Query (class in boto.cloudsearch2.search), 274
 Query (class in boto.sdb.db.query), 925
 query() (boto.dynamodb.layer1.Layer1 method), 330
 query() (boto.dynamodb.layer2.Layer2 method), 334
 query() (boto.dynamodb.table.Table method), 341
 query() (boto.dynamodb2.layer1.DynamoDBConnection method), 386
 query() (boto.dynamodb2.table.Table method), 364
 query() (boto.ec2.cloudwatch.metric.Metric method), 293
 query() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 query_2() (boto.dynamodb2.table.Table method), 365
 query_count() (boto.dynamodb2.table.Table method), 366
 query_gql() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 query_lister() (in module boto.sdb.queryresultset), 742
 query_objects() (boto.datapipeline.layer1.DataPipelineConnection method), 325
 QueryError, 402
 QueryResultSet (class in boto.sdb.queryresultset), 742
 QueryString (boto.gs.connection.GSConnection attribute), 556
 QueryString (boto.s3.connection.S3Connection attribute), 711
 Question (class in boto.mturk.question), 609
 QuestionContent (class in boto.mturk.question), 609
 QuestionForm (class in boto.mturk.question), 609
 QuestionFormAnswer (class in boto.mturk.connection), 605
 Queue (class in boto.sqs.queue), 763
 queue_files() (boto.services.sonofmmm.SonOfMMM method), 744
- ## R
- range_key (boto.dynamodb.item.Item attribute), 345
 range_key_name (boto.dynamodb.item.Item attribute), 345
 range_key_name (boto.dynamodb.schema.Schema attribute), 344
 range_key_type (boto.dynamodb.schema.Schema attribute), 344
 RangeKey (class in boto.dynamodb2.fields), 353
 RankExpressionStatus (class in boto.cloudsearch.optionstatus), 257
 RawMessage (class in boto.sqs.message), 763
 RDSConnection (class in boto.rds), 634
 RDSConnection (class in boto.rds2.layer1), 877
 re_encrypt() (boto.kms.layer1.KMSCConnection method), 827
 read() (boto.connection.HTTPResponse method), 217
 read() (boto.gs.key.Key method), 567
 read() (boto.s3.key.Key method), 722
 read() (boto.sdb.db.blob.Blob method), 918
 read() (boto.sqs.queue.Queue method), 766
 read_job() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491
 read_message() (boto.services.service.Service method), 744
 read_pipeline() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491
 read_preset() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491
 read_units (boto.dynamodb.table.Table attribute), 341
 readline() (boto.sdb.db.blob.Blob method), 918
 ReadReplicaDBInstanceIdentifiers (class in boto.rds.dbinstance), 651
 reboot() (boto.ec2.instance.Instance method), 447
 reboot() (boto.rds.dbinstance.DBInstance method), 650
 reboot_cache_cluster() (boto.elasticache.layer1.ElasticCacheConnection method), 484
 reboot_cluster() (boto.redshift.layer1.RedshiftConnection method), 677
 reboot_db_instance() (boto.rds2.layer1.RDSConnection method), 904
 reboot_dbinstance() (boto.rds.RDSConnection method), 645
 reboot_instance() (boto.opsworks.layer1.OpsWorksConnection method), 864
 reboot_instances() (boto.ec2.connection.EC2Connection method), 430
 rebuild_environment() (boto.beanstalk.layer1.Layer1 method), 198
 RebuildEnvironmentResponse (class in boto.beanstalk.response), 203
 receive_message() (boto.sqs.connection.SQSConnection method), 758
 Record (class in boto.route53.record), 693
 record_activity_task_heartbeat() (boto.swf.layer1.Layer1 method), 785
 record_marker() (boto.swf.layer1_decisions.Layer1Decisions method), 790
 RecurringCharge (class in boto.ec2.reservedinstance), 454
 RedshiftConnection (class in boto.redshift.layer1), 654
 ReferenceProperty (class in boto.sdb.db.property), 924
 refresh() (boto.cloudsearch.optionstatus.OptionStatus method), 257
 refresh() (boto.cloudsearch2.optionstatus.OptionStatus method), 273
 refresh() (boto.dynamodb.table.Table method), 341
 refresh_trusted_advisor_check() (boto.support.layer1.SupportConnection method), 931
 refund() (boto.fps.connection.FPSCConnection method), 520
 refund() (boto.mws.connection.MWSCConnection

- method), 620
- RefundAmount (boto.mws.response.RefundDetails attribute), 629
- RefundDetails (boto.mws.response.RefundResult attribute), 629
- RefundDetails (class in boto.mws.response), 629
- RefundedAmount (boto.mws.response.CaptureDetails attribute), 622
- RefundResult (class in boto.mws.response), 629
- RefundStatus (boto.mws.response.RefundDetails attribute), 629
- RegexConstraint (class in boto.mturk.question), 610
- regions() (in module boto.awslambda), 188
- regions() (in module boto.beanstalk), 192
- regions() (in module boto.cloudformation), 225
- regions() (in module boto.cloudhsm), 243
- regions() (in module boto.cloudsearch), 246
- regions() (in module boto.cloudsearch2), 262
- regions() (in module boto.cloudsearchdomain), 278
- regions() (in module boto.cloudtrail), 283
- regions() (in module boto.codedeploy), 295
- regions() (in module boto.cognito.identity), 306
- regions() (in module boto.cognito.sync), 312
- regions() (in module boto.configservice), 316
- regions() (in module boto.datapipeline), 321
- regions() (in module boto.dynamodb), 327
- regions() (in module boto.dynamodb2), 369
- regions() (in module boto.ec2), 403
- regions() (in module boto.ec2.autoscale), 181
- regions() (in module boto.ec2.cloudwatch), 292
- regions() (in module boto.ec2.elb), 501
- regions() (in module boto.ec2containerservice), 464
- regions() (in module boto.elasticache), 471
- regions() (in module boto.elastictranscoder), 485
- regions() (in module boto.emr), 506
- regions() (in module boto.glacier), 521
- regions() (in module boto.iam), 573
- regions() (in module boto.kinesis), 588
- regions() (in module boto.kms), 821
- regions() (in module boto.logs), 828
- regions() (in module boto.machinelearning), 833
- regions() (in module boto.opsworks), 846
- regions() (in module boto.rds), 647
- regions() (in module boto.rds2), 874
- regions() (in module boto.redshift), 654
- regions() (in module boto.route53.domains), 910
- regions() (in module boto.s3), 697
- regions() (in module boto.sdb), 733
- regions() (in module boto.ses), 745
- regions() (in module boto.sns), 749
- regions() (in module boto.sqs), 754
- regions() (in module boto.sts), 769
- regions() (in module boto.support), 926
- regions() (in module boto.swf), 777
- regions() (in module boto.vpc), 817
- register() (boto.swf.layer2.ActivityType method), 792
- register() (boto.swf.layer2.Domain method), 796
- register() (boto.swf.layer2.WorkflowType method), 799
- register_activity_type() (boto.swf.layer1.Layer1 method), 785
- register_application_revision() (boto.codedeploy.layer1.CodeDeployConnection method), 302
- register_container_instance() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 467
- register_destination() (boto.mws.connection.MWSConnection method), 620
- register_device() (boto.cognito.sync.layer1.CognitoSyncConnection method), 314
- register_domain() (boto.route53.domains.layer1.Route53DomainsConnection method), 912
- register_domain() (boto.swf.layer1.Layer1 method), 786
- register_elastic_ip() (boto.opsworks.layer1.OpsWorksConnection method), 864
- register_hit_type() (boto.mturk.connection.MTurkConnection method), 604
- register_image() (boto.ec2.connection.EC2Connection method), 430
- register_instance() (boto.opsworks.layer1.OpsWorksConnection method), 865
- register_instances() (boto.ec2.elb.ELBConnection method), 500
- register_instances() (boto.ec2.elb.loadbalancer.LoadBalancer method), 505
- register_rds_db_instance() (boto.opsworks.layer1.OpsWorksConnection method), 865
- register_task_definition() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 468
- register_volume() (boto.opsworks.layer1.OpsWorksConnection method), 865
- register_workflow_type() (boto.swf.layer1.Layer1 method), 786
- RegularPrice (boto.mws.response.Offer attribute), 627
- reject_assignment() (boto.mturk.connection.MTurkConnection method), 604
- reject_vpc_peering_connection() (boto.vpc.VPCConnection method), 815
- Relationships (boto.mws.response.Product attribute), 628
- release() (boto.ec2.address.Address method), 404
- release_address() (boto.ec2.connection.EC2Connection method), 431
- remove() (boto.cloudfront.invalidation.InvalidationBatch method), 241
- remove_event_source() (boto.awslambda.layer1.AWSLambdaConnection method), 190

remove_launch_permissions() (boto.ec2.image.Image method), 441
 remove_permission() (boto.sns.SNSConnection method), 753
 remove_permission() (boto.sqs.connection.SQSConnection method), 759
 remove_permission() (boto.sqs.queue.Queue method), 766
 remove_role_from_instance_profile() (boto.iam.connection.IAMConnection method), 585
 remove_rule() (boto.ec2.securitygroup.SecurityGroup method), 457
 remove_source_identifier_from_subscription() (boto.rds2.layer1.RDSConnection method), 904
 remove_tag() (boto.ec2.ec2object.TaggedEC2Object method), 440
 remove_tags() (boto.ec2.ec2object.TaggedEC2Object method), 440
 remove_tags() (boto.emr.connection.EmrConnection method), 509
 remove_tags_from_resource() (boto.rds2.layer1.RDSConnection method), 905
 remove_tags_from_stream() (boto.kinesis.layer1.KinesisConnection method), 595
 remove_user_from_group() (boto.iam.connection.IAMConnection method), 585
 replace_network_acl_entry() (boto.vpc.VPCConnection method), 815
 replace_route() (boto.vpc.VPCConnection method), 816
 replace_route_table_association() (boto.vpc.VPCConnection method), 816
 replace_route_table_association_with_assoc() (boto.vpc.VPCConnection method), 816
 report_task_progress() (boto.datapipeline.layer1.DataPipelineConnection method), 325
 report_task_runner_heartbeat() (boto.datapipeline.layer1.DataPipelineConnection method), 325
 ReportInfo (boto.mws.response.GetReportListResult attribute), 625
 ReportRequestInfo (boto.mws.response.GetReportRequestListResult attribute), 625
 ReportRequestInfo (boto.mws.response.RequestReportResult attribute), 629
 ReportRequestInfo (class in boto.mws.response), 629
 ReportSchedule (boto.mws.response.ManageReportScheduleResult attribute), 627
 Request (class in boto.ec2.autoscale.request), 187
 request_cancel() (boto.swf.layer2.WorkflowExecution method), 797
 request_cancel_activity_task() (boto.swf.layer1_decisions.Layer1Decisions method), 790
 request_cancel_external_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 790
 request_cancel_workflow_execution() (boto.swf.layer1.Layer1 method), 787
 request_environment_info() (boto.beanstalk.layer1.Layer1 method), 199
 request_report() (boto.mws.connection.MWSConnection method), 620
 request_spot_instances() (boto.ec2.connection.EC2Connection method), 431
 RequestEnvironmentInfoResponse (class in boto.beanstalk.response), 203
 RequestHook (class in boto.utils), 222
 RequestReportResult (class in boto.mws.response), 629
 Requirement (class in boto.mturk.qualification), 607
 requires_index_documents (boto.cloudsearch.domain.Domain attribute), 250
 requires_index_documents (boto.cloudsearch2.domain.Domain attribute), 265
 resend_unprocessed() (boto.dynamodb2.table.BatchTable method), 357
 Reservation (class in boto.ec2.instance), 449
 reserve() (boto.fps.connection.FPSConnection method), 520
 ReservedDBInstanceAlreadyExists, 876
 ReservedDBInstanceNotFound, 876
 ReservedDBInstanceQuotaExceeded, 876
 ReservedDBInstancesOfferingNotFound, 876
 ReservedInstance (class in boto.ec2.reservedinstance), 455
 ReservedInstanceListing (class in boto.ec2.reservedinstance), 455
 ReservedInstancesConfiguration (class in boto.ec2.reservedinstance), 455
 ReservedInstancesModification (class in boto.ec2.reservedinstance), 455
 ReservedInstancesOffering (class in boto.ec2.reservedinstance), 456
 ReservedNodeAlreadyExists, 686
 ReservedNodeAlreadyExistsFault, 686
 ReservedNodeNotFound, 687
 ReservedNodeNotFoundFault, 687
 ReservedNodeOfferingNotFound, 687
 ReservedNodeOfferingNotFoundFault, 687
 ReservedNodeQuotaExceeded, 687
 ReservedNodeQuotaExceededFault, 687
 reset_attribute() (boto.ec2.instance.Instance method), 447

[reset_cache_parameter_group\(\)](#)
 (boto.elasticache.layer1.ElastiCacheConnection method), 484

[reset_cluster_parameter_group\(\)](#)
 (boto.redshift.layer1.RedshiftConnection method), 677

[reset_db_parameter_group\(\)](#)
 (boto.rds2.layer1.RDSConnection method), 905

[reset_image_attribute\(\)](#) (boto.ec2.connection.EC2Connection method), 433

[reset_instance_attribute\(\)](#)
 (boto.ec2.connection.EC2Connection method), 433

[reset_launch_attributes\(\)](#) (boto.ec2.image.Image method), 441

[reset_parameter_group\(\)](#) (boto.rds.RDSConnection method), 645

[reset_permissions\(\)](#) (boto.ec2.snapshot.Snapshot method), 458

[reset_snapshot_attribute\(\)](#)
 (boto.ec2.connection.EC2Connection method), 434

[ResizeNotFound](#), 687

[ResizeNotFoundFault](#), 687

[resolve_case\(\)](#) (boto.support.layer1.SupportConnection method), 931

[ResourceAlreadyExistsException](#), 833

[ResourceConflictException](#), 311, 316

[ResourceInUseException](#), 402, 495, 597, 833, 846

[ResourceNotDiscoveredException](#), 320

[ResourceNotFoundException](#), 192, 311, 316, 402, 495, 597, 833, 846, 874

[ResourceRecordBody](#) (boto.route53.record.Record attribute), 693

[ResourceRecordsBody](#) (boto.route53.record.Record attribute), 693

[ResourceRecordSets](#) (class in boto.route53.record), 693

[respond_activity_task_canceled\(\)](#)
 (boto.swf.layer1.Layer1 method), 787

[respond_activity_task_completed\(\)](#)
 (boto.swf.layer1.Layer1 method), 787

[respond_activity_task_failed\(\)](#) (boto.swf.layer1.Layer1 method), 787

[respond_decision_task_completed\(\)](#)
 (boto.swf.layer1.Layer1 method), 788

[response](#) (boto.dynamodb.layer2.TableGenerator attribute), 337

[Response](#) (class in boto.beanstalk.response), 203

[Response](#) (class in boto.mws.response), 629

[ResponseDataElements](#) (boto.glacier.job.Job attribute), 536

[ResponseDataElements](#) (boto.glacier.vault.Vault attribute), 533

[ResponseElement](#) (class in boto.mws.response), 629

[ResponseError](#), 621

[ResponseError](#) (boto.awslambda.layer1.AWSLambdaConnection attribute), 188

[ResponseError](#) (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243

[ResponseError](#) (boto.cloudsearch2.layer1.CloudSearchConnection attribute), 266

[ResponseError](#) (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection attribute), 278

[ResponseError](#) (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284

[ResponseError](#) (boto.codedeploy.layer1.CodeDeployConnection attribute), 296

[ResponseError](#) (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307

[ResponseError](#) (boto.cognito.sync.layer1.CognitoSyncConnection attribute), 312

[ResponseError](#) (boto.configservice.layer1.ConfigServiceConnection attribute), 317

[ResponseError](#) (boto.connection.AWSQueryConnection attribute), 215

[ResponseError](#) (boto.datapipeline.layer1.DataPipelineConnection attribute), 322

[ResponseError](#) (boto.dynamodb.layer1.Layer1 attribute), 328

[ResponseError](#) (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370

[ResponseError](#) (boto.ec2.connection.EC2Connection attribute), 405

[ResponseError](#) (boto.ec2containerservice.layer1.EC2ContainerServiceConnection attribute), 464

[ResponseError](#) (boto.elastictranscoder.layer1.ElasticTranscoderConnection attribute), 485

[ResponseError](#) (boto.emr.connection.EmrConnection attribute), 507

[ResponseError](#) (boto.fps.connection.FPSConnection attribute), 518

[ResponseError](#) (boto.kinesis.layer1.KinesisConnection attribute), 589

[ResponseError](#) (boto.kms.layer1.KMSConnection attribute), 822

[ResponseError](#) (boto.logs.layer1.CloudWatchLogsConnection attribute), 829

[ResponseError](#) (boto.machinelearning.layer1.MachineLearningConnection attribute), 833

[ResponseError](#) (boto.opsworks.layer1.OpsWorksConnection attribute), 847

[ResponseError](#) (boto.rds2.layer1.RDSConnection attribute), 877

[ResponseError](#) (boto.redshift.layer1.RedshiftConnection attribute), 654

[ResponseError](#) (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910

- ResponseError (boto.sdb.connection.SDBConnection attribute), 734
 ResponseError (boto.ses.connection.SESConnection attribute), 745
 ResponseError (boto.sqs.connection.SQSConnection attribute), 755
 ResponseError (boto.support.layer1.SupportConnection attribute), 927
 ResponseError (boto.swf.layer1.Layer1 attribute), 777
 ResponseErrorFactory (class in boto.mws.exception), 621
 ResponseFactory (class in boto.mws.response), 629
 ResponseGroup (class in boto.ecs.item), 470
 ResponseMetadata (boto.mws.response.Response attribute), 629
 ResponseMetadata (class in boto.beanstalk.response), 203
 ResponseResultList (class in boto.mws.response), 629
 restart_app_server() (boto.beanstalk.layer1.Layer1 method), 199
 RestartAppServerResponse (class in boto.beanstalk.response), 203
 restore() (boto.gs.key.Key method), 567
 restore() (boto.s3.key.Key method), 722
 restore_db_instance_from_db_snapshot() (boto.rds2.layer1.RDSConnection method), 905
 restore_db_instance_to_point_in_time() (boto.rds2.layer1.RDSConnection method), 907
 restore_dbinstance_from_dbsnapshot() (boto.rds.RDSConnection method), 646
 restore_dbinstance_from_point_in_time() (boto.rds.RDSConnection method), 646
 restore_from_cluster_snapshot() (boto.redshift.layer1.RedshiftConnection method), 678
 RestoreBody (boto.gs.key.Key attribute), 560
 RestoreBody (boto.s3.key.Key attribute), 716
 resubmit() (boto.dynamodb.batch.BatchList method), 346
 ResultEntry (class in boto.sqs.batchresults), 769
 ResultProcessor (class in boto.services.result), 743
 RESULTS_PER_PAGE (boto.cloudsearch.search.Query attribute), 258
 RESULTS_PER_PAGE (boto.cloudsearch2.search.Query attribute), 274
 ResultSet (class in boto.dynamodb2.results), 356
 ResultSet (class in boto.resultset), 221
 ResumableDownloadException, 219
 ResumableDownloadHandler (class in boto.s3.resumable_download_handler), 729
 ResumableTransferDisposition (class in boto.exception), 219
 ResumableUploadException, 219
 ResumableUploadHandler (class in boto.gs.resumable_upload_handler), 572
 resume_archive_from_file() (boto.glacier.vault.Vault method), 535
 resume_file_upload() (in module boto.glacier.writer), 538
 resume_processes() (boto.ec2.autoscale.AutoScaleConnection method), 180
 resume_processes() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 resync_mfa_device() (boto.iam.connection.IAMConnection method), 585
 retire_grant() (boto.kms.layer1.KMSConnection method), 827
 RetriableResponseError, 621
 retrieve_archive() (boto.glacier.vault.Vault method), 535
 retrieve_domain_auth_code() (boto.route53.domains.layer1.Route53DomainsConnection method), 914
 retrieve_environment_info() (boto.beanstalk.layer1.Layer1 method), 199
 retrieve_inventory() (boto.glacier.vault.Vault method), 535
 retrieve_inventory_job() (boto.glacier.vault.Vault method), 536
 RetrieveEnvironmentInfoResponse (class in boto.beanstalk.response), 203
 retry (boto.mws.exception.ResponseError attribute), 621
 retry (boto.mws.exception.RetriableResponseError attribute), 621
 retry_url() (in module boto.utils), 224
 RETRYABLE_EXCEPTIONS (boto.gs.resumable_upload_handler.ResumableUploadHandler attribute), 572
 RETRYABLE_EXCEPTIONS (boto.s3.resumable_download_handler.ResumableDownloadHandler attribute), 729
 RevisionDoesNotExistException, 306
 RevisionRequiredException, 306
 revoke() (boto.ec2.securitygroup.SecurityGroup method), 457
 revoke() (boto.rds.dbsecuritygroup.DBSecurityGroup method), 652
 revoke_cache_security_group_ingress() (boto.elasticache.layer1.ElastiCacheConnection method), 485
 revoke_cluster_security_group_ingress() (boto.redshift.layer1.RedshiftConnection method), 680
 revoke_db_security_group_ingress() (boto.rds2.layer1.RDSConnection method), 909
 revoke_dbsecurity_group() (boto.rds.RDSConnection method), 647
 revoke_grant() (boto.kms.layer1.KMSConnection method), 827

revoke_qualification() (boto.mturk.connection.MTurkConnection method), 604
 revoke_security_group() (boto.ec2.connection.EC2Connection method), 434
 revoke_security_group() (boto.rds.RDSConnection method), 647
 revoke_security_group_deprecated() (boto.ec2.connection.EC2Connection method), 434
 revoke_security_group_egress() (boto.ec2.connection.EC2Connection method), 435
 revoke_snapshot_access() (boto.redshift.layer1.RedshiftConnection method), 680
 RoleRequiredException, 306
 rotate_encryption_key() (boto.redshift.layer1.RedshiftConnection method), 681
 Route (class in boto.vpc.routetable), 818
 Route53Connection (class in boto.route53.connection), 688
 Route53DomainsConnection (class in boto.route53.domains.layer1), 910
 RouteAssociation (class in boto.vpc.routetable), 818
 RouteTable (class in boto.vpc.routetable), 818
 RRRBody (boto.route53.record.Record attribute), 693
 Rule (class in boto.s3.lifecycle), 731
 run() (boto.ec2.image.Image method), 441
 run() (boto.glacier.concurrent.TransferThread method), 539
 run() (boto.manage.cmdshell.LocalClient method), 598
 run() (boto.manage.cmdshell.SSHClient method), 599
 run() (boto.pyami.scriptbase.ScriptBase method), 633
 run() (boto.sdb.connection.ItemThread method), 733
 run() (boto.sdb.domain.UploaderThread method), 741
 run() (boto.utils.ShellCommand method), 222
 run_instances() (boto.ec2.connection.EC2Connection method), 435
 run_jobflow() (boto.emr.connection.EmrConnection method), 509
 run_pty() (boto.manage.cmdshell.SSHClient method), 599
 run_scripts() (boto.pyami.startup.Startup method), 633
 run_task() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 468
 RunJobFlowResponse (class in boto.emr.emrobject), 515

S

S3BucketDoesNotExistException, 287
 S3Connection (class in boto.s3.connection), 711
 S3CopyError, 219
 S3CreateError, 219
 S3DataError, 219
 S3KeyProperty (class in boto.sdb.db.property), 925
 S3Location (class in boto.beanstalk.response), 203
 S3Origin (class in boto.cloudfront.origin), 240
 S3PermissionsError, 219
 S3RegionInfo (class in boto.s3), 697
 S3ResponseError, 219
 S3WebsiteEndpointTranslate (class in boto.s3.bucket), 709
 SAEast (boto.s3.connection.Location attribute), 711
 SalePrice (boto.mws.response.CartItem attribute), 622
 SalesRank (class in boto.mws.response), 629
 SalesRankings (boto.mws.response.Product attribute), 628
 save() (boto.cloudsearch.optionstatus.IndexFieldStatus method), 257
 save() (boto.cloudsearch.optionstatus.OptionStatus method), 257
 save() (boto.cloudsearch2.optionstatus.AvailabilityOptionsStatus method), 272
 save() (boto.cloudsearch2.optionstatus.IndexFieldStatus method), 272
 save() (boto.cloudsearch2.optionstatus.OptionStatus method), 273
 save() (boto.dynamodb.item.Item method), 345
 save() (boto.dynamodb2.items.Item method), 355
 save() (boto.ec2.keypair.KeyPair method), 451
 save() (boto.sdb.item.Item method), 742
 save() (boto.sqs.queue.Queue method), 766
 save() (boto.sts.credentials.Credentials method), 776
 save_object() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 save_option() (boto.pyami.config.Config method), 631
 save_results() (boto.services.service.Service method), 744
 save_system_option() (boto.pyami.config.Config method), 631
 save_to_file() (boto.sqs.queue.Queue method), 766
 save_to_filename() (boto.sqs.queue.Queue method), 766
 save_to_s3() (boto.sqs.queue.Queue method), 766
 save_user_option() (boto.pyami.config.Config method), 631
 SavedCartItem (boto.mws.response.Cart attribute), 622
 ScalingParametersStatus (class in boto.cloudsearch2.optionstatus), 273
 ScalingPolicy (class in boto.ec2.autoscale.policy), 186
 scan() (boto.dynamodb.layer1.Layer1 method), 330
 scan() (boto.dynamodb.layer2.Layer2 method), 335
 scan() (boto.dynamodb.table.Table method), 342
 scan() (boto.dynamodb2.layer1.DynamoDBConnection method), 391
 scan() (boto.dynamodb2.table.Table method), 367
 scanned_count (boto.dynamodb.layer2.TableGenerator attribute), 337
 schedule_activity_task() (boto.swf.layer1_decisions.Layer1Decisions

method), 791

ScheduledUpdateGroupAction (class in boto.ec2.autoscale.scheduled), 187

schema (boto.dynamodb.table.Table attribute), 343

Schema (class in boto.dynamodb.schema), 343

schema() (boto.dynamodb2.fields.BaseIndexField method), 349

schema() (boto.dynamodb2.fields.BaseSchemaField method), 350

schema() (boto.dynamodb2.fields.GlobalBaseIndexField method), 350

schema() (boto.dynamodb2.fields.GlobalIncludeIndex method), 351

schema() (boto.dynamodb2.fields.IncludeIndex method), 352

schema_url (boto.mturk.question.ExternalQuestion attribute), 608

schema_url (boto.mturk.question.FormattedContent attribute), 608

schema_url (boto.mturk.question.HTMLQuestion attribute), 608

schema_url (boto.mturk.question.QuestionForm attribute), 610

Scope (class in boto.gs.acl), 540

ScriptBase (class in boto.pyami.scriptbase), 633

ScriptRunnerJar (boto.emr.step.ScriptRunnerStep attribute), 511

ScriptRunnerStep (class in boto.emr.step), 511

sdb (boto.sdb.db.manager.sdbmanager.SDBManager attribute), 921

SDBConnection (class in boto.sdb.connection), 733

SDBConverter (class in boto.sdb.db.manager.sdbmanager), 919

SDBManager (class in boto.sdb.db.manager.sdbmanager), 920

SDBPersistenceError, 219

SDBResponseError, 219

search() (boto.cloudsearch.search.SearchConnection method), 259

search() (boto.cloudsearch2.search.SearchConnection method), 275

search() (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection method), 278

search_hits() (boto.mturk.connection.MTurkConnection method), 604

search_instance_count (boto.cloudsearch.domain.Domain attribute), 250

search_instance_count (boto.cloudsearch2.domain.Domain attribute), 265

search_partition_count (boto.cloudsearch.domain.Domain attribute), 250

search_partition_count (boto.cloudsearch2.domain.Domain attribute), 265

search_qualification_types() (boto.mturk.connection.MTurkConnection method), 604

search_scopes() (boto.mws.connection.MWSConnection.ResponseFactory method), 611

search_scopes() (boto.mws.response.ResponseFactory method), 629

search_service_arn (boto.cloudsearch.domain.Domain attribute), 250

search_service_endpoint (boto.cloudsearch.domain.Domain attribute), 250

search_service_endpoint (boto.cloudsearch2.domain.Domain attribute), 265

SearchConnection (class in boto.cloudsearch.search), 258

SearchConnection (class in boto.cloudsearch2.search), 274

SearchException, 283

SearchResults (class in boto.cloudsearch.search), 260

SearchResults (class in boto.cloudsearch2.search), 276

SearchServiceException, 260, 262, 276, 277

secret_key (boto.connection.AWSAuthConnection attribute), 215

secret_key (boto.gs.connection.GSConnection attribute), 558

security_group (boto.rds.dbinstance.DBInstance attribute), 650

SecurityGroup (class in boto.ec2.elb.securitygroup), 505

SecurityGroup (class in boto.ec2.securitygroup), 456

select() (boto.sdb.connection.SDBConnection method), 738

select() (boto.sdb.domain.Domain method), 740

select_listener() (in module boto.sdb.queryresultset), 742

SELECTION_VALUE_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610

SELECTION_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610

SelectionAnswer (class in boto.mturk.question), 610

SELECTIONANSWER_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610

SelectionResultSet (class in boto.sdb.queryresultset), 742

Self (boto.mws.response.GetProductCategoriesResult attribute), 625

SellerOrderAttributes (boto.mws.response.OrderReferenceDetails attribute), 628

send_email() (boto.ses.connection.SESConnection method), 746

send_file() (boto.gs.key.Key method), 567

send_file() (boto.gs.resumable_upload_handler.ResumableUploadHandler method), 572

send_file() (boto.s3.key.Key method), 723

send_message() (boto.sqs.connection.SQSConnection method), 759

send_message_batch() (boto.sqs.connection.SQSConnection method), 759
 send_raw_email() (boto.ses.connection.SESConnection method), 747
 send_test_event_notification() (boto.mturk.connection.MTurkConnection method), 604
 send_test_notification_to_destination() (boto.mws.connection.MWSConnection method), 620
 serialize_num() (in module boto.dynamodb.types), 348
 SERVER_HAS_NOTHING (boto.gs.resumable_upload_handler.ResumableUploadHandler.item.ResponseGroup attribute), 572
 server_name() (boto.connection.AWSAuthConnection method), 215
 server_name() (boto.gs.connection.GSConnection method), 559
 ServerException, 469
 Service (class in boto.services.service), 744
 service_arn (boto.cloudsearch2.domain.Domain attribute), 266
 SERVICE_NAME (boto.mturk.notification.NotificationMessage attribute), 606
 ServiceDef (class in boto.services.servicedef), 744
 ServiceException, 192
 ServiceMessage (class in boto.services.message), 743
 ServiceName (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243
 ServiceName (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284
 ServiceName (boto.codedeploy.layer1.CodeDeployConnection attribute), 296
 ServiceName (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307
 ServiceName (boto.configservice.layer1.ConfigServiceConnection attribute), 317
 ServiceName (boto.datapipeline.layer1.DataPipelineConnection attribute), 322
 ServiceName (boto.dynamodb.layer1.Layer1 attribute), 328
 ServiceName (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370
 ServiceName (boto.kinesis.layer1.KinesisConnection attribute), 589
 ServiceName (boto.kms.layer1.KMSConnection attribute), 822
 ServiceName (boto.logs.layer1.CloudWatchLogsConnection attribute), 829
 ServiceName (boto.machinelearning.layer1.MachineLearningConnection attribute), 833
 ServiceName (boto.opsworks.layer1.OpsWorksConnection attribute), 847
 ServiceName (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910
 ServiceName (boto.support.layer1.SupportConnection attribute), 927
 ServiceName (boto.swf.layer1.Layer1 attribute), 777
 ServicePoliciesStatus (class in boto.cloudsearch.optionstatus), 257
 ServicePoliciesStatus (class in boto.cloudsearch2.optionstatus), 273
 ServiceUnavailableException, 833
 SESConnection (class in boto.ses.connection), 745
 SessionExpiredError (boto.dynamodb.layer1.Layer1 attribute), 328
 set() (boto.ec2.elastic.item.ResponseGroup method), 470
 set() (boto.utils.Password method), 222
 set_acl() (boto.gs.bucket.Bucket method), 552
 set_acl() (boto.gs.key.Key method), 567
 set_acl() (boto.s3.bucket.Bucket method), 708
 set_acl() (boto.s3.key.Key method), 723
 set_alarm_state() (boto.ec2.cloudwatch.CloudWatchConnection method), 292
 set_alias() (boto.route53.record.Record method), 693
 set_as_logging_target() (boto.gs.bucket.Bucket method), 553
 set_as_logging_target() (boto.s3.bucket.Bucket method), 708
 set_attribute() (boto.sqs.queue.Queue method), 766
 set_body() (boto.contrib.ymlmessage.YAMLMessage method), 321
 set_body() (boto.sqs.message.RawMessage method), 763
 set_bucket_class() (boto.gs.connection.GSConnection method), 559
 set_bucket_class() (boto.s3.connection.S3Connection method), 714
 set_canned_acl() (boto.gs.bucket.Bucket method), 553
 set_canned_acl() (boto.gs.key.Key method), 568
 set_canned_acl() (boto.s3.bucket.Bucket method), 708
 set_canned_acl() (boto.s3.key.Key method), 723
 set_capacity() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 set_contents_from_file() (boto.file.key.Key method), 517
 set_contents_from_file() (boto.gs.key.Key method), 568
 set_contents_from_file() (boto.s3.key.Key method), 723
 set_contents_from_filename() (boto.gs.key.Key method), 569
 set_contents_from_filename() (boto.s3.key.Key method), 724
 set_contents_from_stream() (boto.gs.key.Key method), 570
 set_contents_from_stream() (boto.s3.key.Key method), 725
 set_contents_from_string() (boto.gs.key.Key method), 571
 set_contents_from_string() (boto.s3.key.Key method), 726

- set_cors() (boto.gs.bucket.Bucket method), 553
 set_cors() (boto.s3.bucket.Bucket method), 708
 set_cors_xml() (boto.gs.bucket.Bucket method), 553
 set_cors_xml() (boto.s3.bucket.Bucket method), 708
 set_def_acl() (boto.gs.bucket.Bucket method), 553
 set_def_canned_acl() (boto.gs.bucket.Bucket method), 553
 set_def_xml_acl() (boto.gs.bucket.Bucket method), 554
 set_default_credentials() (in module boto.swf.layer2), 801
 set_default_policy_version() (boto.iam.connection.IAMConnection method), 585
 set_desired_capacity() (boto.ec2.autoscale.AutoScaleConnection method), 180
 set_distribution_config() (boto.cloudfront.CloudFrontConnection method), 235
 set_email_notification() (boto.mturk.connection.MTurkConnection method), 604
 set_encryption_config() (boto.gs.bucket.Bucket method), 554
 set_endpoint_attributes() (boto.sns.SNSConnection method), 753
 set_file_logger() (in module boto), 213
 set_host_header() (boto.connection.AWSAuthConnection method), 215
 set_host_header() (boto.gs.connection.GSConnection method), 559
 set_identity_dkim_enabled() (boto.ses.connection.SESConnection method), 747
 set_identity_feedback_forwarding_enabled() (boto.ses.connection.SESConnection method), 748
 set_identity_notification_topic() (boto.ses.connection.SESConnection method), 748
 set_identity_pool_configuration() (boto.cognito.sync.layer1.CognitoSyncConnection method), 314
 set_instance_health() (boto.ec2.autoscale.AutoScaleConnection method), 181
 set_item_cls() (boto.sdb.connection.SDBConnection method), 738
 set_key_class() (boto.gs.bucket.Bucket method), 554
 set_key_class() (boto.s3.bucket.Bucket method), 708
 set_key_value() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 set_launch_permissions() (boto.ec2.image.Image method), 444
 set_lb_listener_SSL_certificate() (boto.ec2.elb.ELBConnection method), 500
 set_lb_policies_of_backend_server() (boto.ec2.elb.ELBConnection method), 500
 set_lb_policies_of_listener() (boto.ec2.elb.ELBConnection method), 501
 set_listener_SSL_certificate() (boto.ec2.elb.loadbalancer.LoadBalancer method), 505
 set_load_based_auto_scaling() (boto.opsworks.layer1.OpsWorksConnection method), 865
 set_message_class() (boto.sqs.queue.Queue method), 768
 set_metadata() (boto.gs.key.Key method), 571
 set_metadata() (boto.s3.key.Key method), 726
 set_next_token() (boto.sdb.db.query.Query method), 926
 set_order_reference_details() (boto.mws.connection.MWSConnection method), 620
 set_origin_access_identity_config() (boto.cloudfront.CloudFrontConnection method), 235
 set_permission() (boto.opsworks.layer1.OpsWorksConnection method), 866
 set_permissions() (boto.cloudfront.distribution.Distribution method), 237
 set_permissions_all() (boto.cloudfront.distribution.Distribution method), 237
 set_platform_application_attributes() (boto.sns.SNSConnection method), 753
 set_policies_of_backend_server() (boto.ec2.elb.loadbalancer.LoadBalancer method), 505
 set_policies_of_listener() (boto.ec2.elb.loadbalancer.LoadBalancer method), 505
 set_policy() (boto.cloudformation.stack.Stack method), 233
 set_policy() (boto.gs.bucket.Bucket method), 554
 set_policy() (boto.s3.bucket.Bucket method), 708
 set_property() (boto.sdb.db.manager.sdbmanager.SDBManager method), 921
 set_queue_attribute() (boto.sqs.connection.SQSConnection method), 760
 set_redirect() (boto.gs.key.Key method), 571
 set_redirect() (boto.s3.key.Key method), 726
 set_remote_metadata() (boto.gs.key.Key method), 571
 set_remote_metadata() (boto.s3.key.Key method), 726
 set_request_hook() (boto.connection.AWSAuthConnection method), 215
 set_request_hook() (boto.gs.connection.GSConnection method), 559
 set_request_payment() (boto.gs.bucket.Bucket method), 554
 set_request_payment() (boto.s3.bucket.Bucket method), 708
 set_rest_notification() (boto.mturk.connection.MTurkConnection method), 604

set_retention() (boto.logs.layer1.CloudWatchLogsConnection method), 832
 set_reviewing() (boto.mturk.connection.MTurkConnection method), 604
 set_sqs_notification() (boto.mturk.connection.MTurkConnection method), 604
 set_stack_policy() (boto.cloudformation.connection.CloudFormationConnection method), 230
 set_state() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
 set_status() (boto.datapipeline.layer1.DataPipelineConnection method), 326
 set_storage_class() (boto.gs.bucket.Bucket method), 554
 set_stream_logger() (in module boto), 213
 set_streaming_distribution_config() (boto.cloudfront.CloudFrontConnection method), 235
 set_subresource() (boto.gs.bucket.Bucket method), 554
 set_subresource() (boto.s3.bucket.Bucket method), 709
 set_tags() (boto.gs.bucket.Bucket method), 554
 set_tags() (boto.s3.bucket.Bucket method), 709
 set_task_status() (boto.datapipeline.layer1.DataPipelineConnection method), 326
 set_termination_protection() (boto.emr.connection.EmrConnection method), 510
 set_time_based_auto_scaling() (boto.opsworks.layer1.OpsWorksConnection method), 866
 set_timeout() (boto.sqs.queue.Queue method), 768
 set_topic_attributes() (boto.sns.SNSConnection method), 753
 set_value() (boto.rds.parametergroup.Parameter method), 653
 set_vault_notifications() (boto.glacier.layer1.Layer1 method), 530
 set_visible_to_all_users() (boto.emr.connection.EmrConnection method), 510
 set_website_configuration() (boto.gs.bucket.Bucket method), 554
 set_website_configuration() (boto.s3.bucket.Bucket method), 709
 set_website_configuration_xml() (boto.gs.bucket.Bucket method), 554
 set_website_configuration_xml() (boto.s3.bucket.Bucket method), 709
 set_xml_acl() (boto.gs.bucket.Bucket method), 554
 set_xml_acl() (boto.gs.key.Key method), 571
 set_xml_acl() (boto.s3.bucket.Bucket method), 709
 set_xml_acl() (boto.s3.key.Key method), 726
 set_xml_logging() (boto.gs.bucket.Bucket method), 555
 set_xml_logging() (boto.s3.bucket.Bucket method), 709
 set_xml_tags() (boto.gs.bucket.Bucket method), 555
 set_xml_tags() (boto.s3.bucket.Bucket method), 709
 setlocale() (in module boto.utils), 224
 SetOrderReferenceDetailsResult (class in boto.mws.response), 629
 setReadOnly() (boto.utils.ShellCommand method), 222
 setup() (boto.cloudformation.connection.FPSConnection method), 520
 settle_debt() (boto.fps.connection.FPSConnection method), 520
 setup() (boto.mws.response.DeclarativeType method), 623
 setup_vhost() (boto.pyami.installers.ubuntu.trac.Trac method), 633
 share() (boto.ec2.snapshot.Snapshot method), 458
 shell() (boto.manage.cmdshell.LocalClient method), 598
 shell() (boto.manage.cmdshell.SSHClient method), 599
 ShellCommand (class in boto.utils), 222
 ShipmentData (boto.mws.response.ListInboundShipmentsResult attribute), 626
 Shipping (boto.mws.response.Price attribute), 628
 ShippingAddress (boto.mws.response.Order attribute), 627
 ShippingAddressList (boto.mws.response.Customer attribute), 623
 ShippingDiscount (boto.mws.response.OrderItem attribute), 628
 ShippingPrice (boto.mws.response.OrderItem attribute), 628
 ShippingTax (boto.mws.response.OrderItem attribute), 628
 ShipToAddress (boto.mws.response.GetPackageTrackingDetailsResult attribute), 625
 should_flush() (boto.dynamodb2.table.BatchTable method), 357
 should_retry() (boto.gs.key.Key method), 572
 should_retry() (boto.s3.key.Key method), 726
 shutdown() (boto.services.service.Service method), 744
 shutdown() (boto.services.sonofmmm.SonOfMMM method), 744
 shutdown_instances() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 signal() (boto.swf.layer2.WorkflowExecution method), 798
 signal_external_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 791
 signal_workflow_execution() (boto.swf.layer1.Layer1 method), 788
 Signer (class in boto.cloudfront.signers), 241
 SimpleField (class in boto.mturk.question), 610
 SimpleList (class in boto.mws.response), 629
 SimpleResultSet (class in boto.file.simpleresultset), 516
 SingleOperationThreshold (boto.glacier.vault.Vault at-

- tribute), 533
- size (boto.sdb.db.blob.Blob attribute), 918
- size() (boto.connection.ConnectionPool method), 216
- size() (boto.connection.HostConnectionPool method), 217
- size_bytes (boto.dynamodb.table.Table attribute), 343
- skip_proxy() (boto.connection.AWSAuthConnection method), 215
- skip_proxy() (boto.gs.connection.GSConnection method), 559
- SmallImage (boto.mws.response.AttributeSet attribute), 621
- Snapshot (class in boto.ec2.snapshot), 457
- snapshot() (boto.rds.dbinstance.DBInstance method), 650
- SnapshotAttribute (class in boto.ec2.snapshot), 458
- SnapshotCopyAlreadyDisabled, 687
- SnapshotCopyAlreadyEnabled, 687
- SnapshotCopyDisabled, 687
- SnapshotQuotaExceeded, 876
- snapshots() (boto.ec2.volume.Volume method), 461
- SNSConnection (class in boto.sns), 749
- SNSInvalidTopic, 687, 876
- SNSNoAuthorization, 687, 876
- SNSTopicArnNotFound, 687, 876
- SolutionStackDescription (class in boto.beanstalk.response), 203
- SonOfMMM (class in boto.services.sonofmmm), 744
- SourceNotFound, 687, 876
- split_key() (boto.services.service.Service method), 744
- split_shard() (boto.kinesis.layer1.KinesisConnection method), 596
- SpotInstanceRequest (class in boto.ec2.spotinstancerequest), 458
- SpotInstanceStateFault (class in boto.ec2.spotinstancerequest), 459
- SpotInstanceStatus (class in boto.ec2.spotinstancerequest), 459
- SQSConnection (class in boto.sqs.connection), 755
- SQSDecodeError, 220
- SQSError, 220
- SQSRegionInfo (class in boto.sqs.regioninfo), 768
- SSHClient (class in boto.manage.cmdshell), 598
- sshclient_from_instance() (in module boto.manage.cmdshell), 599
- Stack (class in boto.cloudformation.stack), 233
- stack_name_reason (boto.cloudformation.stack.Stack attribute), 233
- StackEvent (class in boto.cloudformation.stack), 233
- StackResource (class in boto.cloudformation.stack), 233
- StackResourceSummary (class in boto.cloudformation.stack), 233
- StackSummary (class in boto.cloudformation.stack), 233
- STALE_DURATION (boto.connection.ConnectionPool attribute), 216
- start() (boto.ec2.instance.Instance method), 447
- start() (boto.mws.response.DeclarativeType method), 623
- start() (boto.mws.response.Element method), 623
- start() (boto.mws.response.ElementList method), 624
- start() (boto.mws.response.SimpleList method), 629
- start() (boto.pyami.installers.Installer method), 632
- start() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- start() (boto.swf.layer2.WorkflowType method), 800
- start() (in module boto.manage.cmdshell), 600
- start_child_workflow_execution() (boto.swf.layer1_decisions.Layer1Decisions method), 791
- start_configuration_recorder() (boto.configservice.layer1.ConfigServiceConnection method), 319
- start_instance() (boto.opsworks.layer1.OpsWorksConnection method), 867
- start_instances() (boto.ec2.connection.EC2Connection method), 438
- start_logging() (boto.cloudtrail.layer1.CloudTrailConnection method), 285
- START_OVER (boto.exception.ResumableTransferDisposition attribute), 219
- start_stack() (boto.opsworks.layer1.OpsWorksConnection method), 867
- start_task() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 468
- start_timer() (boto.swf.layer1_decisions.Layer1Decisions method), 791
- start_workflow_execution() (boto.swf.layer1.Layer1 method), 788
- startElement() (boto.cloudformation.stack.Capability method), 232
- startElement() (boto.cloudformation.stack.NotificationARN method), 232
- startElement() (boto.cloudformation.stack.Output method), 232
- startElement() (boto.cloudformation.stack.Parameter method), 233
- startElement() (boto.cloudformation.stack.Stack method), 233
- startElement() (boto.cloudformation.stack.StackEvent method), 233
- startElement() (boto.cloudformation.stack.StackResource method), 233
- startElement() (boto.cloudformation.stack.StackResourceSummary method), 233
- startElement() (boto.cloudformation.stack.StackSummary method), 234
- startElement() (boto.cloudformation.stack.Tag method), 234
- startElement() (boto.cloudformation.template.Template method), 234

startElement() (boto.cloudformation.template.TemplateParameter method), 234

startElement() (boto.cloudfront.distribution.Distribution method), 237

startElement() (boto.cloudfront.distribution.DistributionConfig method), 238

startElement() (boto.cloudfront.distribution.DistributionSummary method), 238

startElement() (boto.cloudfront.distribution.StreamingDistribution method), 238

startElement() (boto.cloudfront.identity.OriginAccessIdentity method), 240

startElement() (boto.cloudfront.identity.OriginAccessIdentityConfig method), 240

startElement() (boto.cloudfront.identity.OriginAccessIdentitySummary method), 241

startElement() (boto.cloudfront.invalidation.InvalidationBatch method), 241

startElement() (boto.cloudfront.invalidation.InvalidationList method), 242

startElement() (boto.cloudfront.invalidation.InvalidationSummary method), 242

startElement() (boto.cloudfront.logging.LoggingInfo method), 243

startElement() (boto.cloudfront.origin.CustomOrigin method), 240

startElement() (boto.cloudfront.origin.S3Origin method), 240

startElement() (boto.cloudfront.signers.ActiveTrustedSigners method), 241

startElement() (boto.cloudfront.signers.Signer method), 241

startElement() (boto.cloudfront.signers.TrustedSigners method), 241

startElement() (boto.cloudsearch.optionstatus.OptionStatus method), 257

startElement() (boto.ec2.autoscale.activity.Activity method), 182

startElement() (boto.ec2.autoscale.group.AutoScalingGroup method), 183

startElement() (boto.ec2.autoscale.group.AutoScalingGroupMetric method), 183

startElement() (boto.ec2.autoscale.group.EnabledMetric method), 183

startElement() (boto.ec2.autoscale.group.ProcessType method), 183

startElement() (boto.ec2.autoscale.group.SuspendedProcess method), 184

startElement() (boto.ec2.autoscale.group.TerminationPolicies method), 184

startElement() (boto.ec2.autoscale.instance.Instance method), 184

startElement() (boto.ec2.autoscale.launchconfig.BlockDeviceMapping method), 184

startElement() (boto.ec2.autoscale.launchconfig.Ebs method), 184

startElement() (boto.ec2.autoscale.launchconfig.InstanceMonitoring method), 184

startElement() (boto.ec2.autoscale.launchconfig.LaunchConfiguration method), 186

startElement() (boto.ec2.autoscale.policy.AdjustmentType method), 186

startElement() (boto.ec2.autoscale.policy.Alarm method), 186

startElement() (boto.ec2.autoscale.policy.MetricCollectionTypes method), 186

startElement() (boto.ec2.autoscale.policy.MetricCollectionTypes.BaseType method), 186

startElement() (boto.ec2.autoscale.policy.ScalingPolicy method), 187

startElement() (boto.ec2.autoscale.policy.TerminationPolicies method), 187

startElement() (boto.ec2.autoscale.request.Request method), 187

startElement() (boto.ec2.autoscale.scheduled.ScheduledUpdateGroupAction method), 187

startElement() (boto.ec2.autoscale.tag.Tag method), 188

startElement() (boto.ec2.blockdevicemapping.BlockDeviceMapping method), 404

startElement() (boto.ec2.blockdevicemapping.BlockDeviceType method), 404

startElement() (boto.ec2.cloudwatch.alarm.AlarmHistoryItem method), 294

startElement() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295

startElement() (boto.ec2.cloudwatch.alarm.MetricAlarms method), 295

startElement() (boto.ec2.cloudwatch.datapoint.Datapoint method), 292

startElement() (boto.ec2.cloudwatch.metric.Metric method), 293

startElement() (boto.ec2.ec2object.EC2Object method), 440

startElement() (boto.ec2.ec2object.TaggedEC2Object method), 441

startElement() (boto.ec2.elb.attributes.AccessLogAttribute method), 506

startElement() (boto.ec2.elb.attributes.ConnectionDrainingAttribute method), 506

startElement() (boto.ec2.elb.attributes.ConnectionSettingAttribute method), 506

startElement() (boto.ec2.elb.attributes.CrossZoneLoadBalancingAttribute method), 506

startElement() (boto.ec2.elb.attributes.LbAttributes method), 506

startElement() (boto.ec2.elb.healthcheck.HealthCheck method), 501

startElement() (boto.ec2.elb.instancestate.InstanceState method), 501

- method), 502
- startElement() (boto.ec2.elb.listelement.ListElement method), 502
- startElement() (boto.ec2.elb.listener.Listener method), 502
- startElement() (boto.ec2.elb.loadbalancer.Backend method), 502
- startElement() (boto.ec2.elb.loadbalancer.LoadBalancer method), 505
- startElement() (boto.ec2.elb.loadbalancer.LoadBalancerZones method), 505
- startElement() (boto.ec2.elb.policies.AppCookieStickinessPolicy method), 505
- startElement() (boto.ec2.elb.policies.LBCookieStickinessPolicy method), 505
- startElement() (boto.ec2.elb.policies.OtherPolicy method), 505
- startElement() (boto.ec2.elb.policies.Policies method), 505
- startElement() (boto.ec2.elb.securitygroup.SecurityGroup method), 505
- startElement() (boto.ec2.group.Group method), 441
- startElement() (boto.ec2.image.BillingProducts method), 441
- startElement() (boto.ec2.image.CopyImage method), 441
- startElement() (boto.ec2.image.Image method), 444
- startElement() (boto.ec2.image.ImageAttribute method), 444
- startElement() (boto.ec2.image.ProductCodes method), 444
- startElement() (boto.ec2.instance.ConsoleOutput method), 444
- startElement() (boto.ec2.instance.Instance method), 447
- startElement() (boto.ec2.instance.InstanceAttribute method), 448
- startElement() (boto.ec2.instance.InstancePlacement method), 448
- startElement() (boto.ec2.instance.InstanceState method), 449
- startElement() (boto.ec2.instance.Reservation method), 449
- startElement() (boto.ec2.instance.SubParse method), 449
- startElement() (boto.ec2.instanceinfo.InstanceInfo method), 449
- startElement() (boto.ec2.instancestatus.Details method), 450
- startElement() (boto.ec2.instancestatus.Event method), 450
- startElement() (boto.ec2.instancestatus.EventSet method), 450
- startElement() (boto.ec2.instancestatus.InstanceStatus method), 450
- startElement() (boto.ec2.instancestatus.InstanceStatusSet method), 450
- startElement() (boto.ec2.instancestatus.Status method), 451
- startElement() (boto.ec2.launchspecification.GroupList method), 452
- startElement() (boto.ec2.launchspecification.LaunchSpecification method), 452
- startElement() (boto.ec2.networkinterface.Attachment method), 452
- startElement() (boto.ec2.networkinterface.NetworkInterface method), 453
- startElement() (boto.ec2.networkinterface.PrivateIPAddress method), 453
- startElement() (boto.ec2.reservedinstance.InstanceCount method), 454
- startElement() (boto.ec2.reservedinstance.ModificationResult method), 454
- startElement() (boto.ec2.reservedinstance.ModifyReservedInstancesResult method), 454
- startElement() (boto.ec2.reservedinstance.PriceSchedule method), 454
- startElement() (boto.ec2.reservedinstance.PricingDetail method), 454
- startElement() (boto.ec2.reservedinstance.RecurringCharge method), 455
- startElement() (boto.ec2.reservedinstance.ReservedInstanceListing method), 455
- startElement() (boto.ec2.reservedinstance.ReservedInstancesConfiguration method), 455
- startElement() (boto.ec2.reservedinstance.ReservedInstancesModification method), 455
- startElement() (boto.ec2.reservedinstance.ReservedInstancesOffering method), 456
- startElement() (boto.ec2.securitygroup.GroupOrCIDR method), 456
- startElement() (boto.ec2.securitygroup.IPPermissions method), 456
- startElement() (boto.ec2.securitygroup.IPPermissionsList method), 456
- startElement() (boto.ec2.securitygroup.SecurityGroup method), 457
- startElement() (boto.ec2.snapshot.SnapshotAttribute method), 458
- startElement() (boto.ec2.spotinstancerequest.SpotInstanceRequest method), 459
- startElement() (boto.ec2.spotinstancerequest.SpotInstanceStateFault method), 459
- startElement() (boto.ec2.spotinstancerequest.SpotInstanceStatus method), 459
- startElement() (boto.ec2.tag.Tag method), 460
- startElement() (boto.ec2.tag.TagSet method), 460
- startElement() (boto.ec2.volume.AttachmentSet method), 460
- startElement() (boto.ec2.volume.Volume method), 462
- startElement() (boto.ec2.volume.VolumeAttribute

method), 462
startElement() (boto.ec2.volumestatus.Action method), 462
startElement() (boto.ec2.volumestatus.ActionSet method), 462
startElement() (boto.ec2.volumestatus.Event method), 462
startElement() (boto.ec2.volumestatus.EventSet method), 463
startElement() (boto.ec2.volumestatus.VolumeStatus method), 463
startElement() (boto.ec2.volumestatus.VolumeStatusSet method), 463
startElement() (boto.ec2.zone.MessageSet method), 463
startElement() (boto.ec2.zone.Zone method), 463
startElement() (boto.ecs.item.ItemSet method), 470
startElement() (boto.ecs.item.ResponseGroup method), 471
startElement() (boto.emr.emrobject.BootstrapAction method), 513
startElement() (boto.emr.emrobject.BootstrapActionList method), 513
startElement() (boto.emr.emrobject.Cluster method), 513
startElement() (boto.emr.emrobject.ClusterStatus method), 513
startElement() (boto.emr.emrobject.ClusterSummary method), 513
startElement() (boto.emr.emrobject.ClusterSummaryList method), 513
startElement() (boto.emr.emrobject.EmrObject method), 514
startElement() (boto.emr.emrobject.HadoopStep method), 514
startElement() (boto.emr.emrobject.InstanceGroupInfo method), 514
startElement() (boto.emr.emrobject.InstanceGroupList method), 514
startElement() (boto.emr.emrobject.InstanceInfo method), 514
startElement() (boto.emr.emrobject.InstanceList method), 514
startElement() (boto.emr.emrobject.JobFlow method), 514
startElement() (boto.emr.emrobject.JobFlowStepList method), 514
startElement() (boto.emr.emrobject.Step method), 515
startElement() (boto.emr.emrobject.StepConfig method), 515
startElement() (boto.emr.emrobject.StepSummary method), 515
startElement() (boto.emr.emrobject.StepSummaryList method), 515
startElement() (boto.exception.BotoServerError method), 218
startElement() (boto.exception.ConsoleOutput method), 218
startElement() (boto.exception.EC2ResponseError method), 218
startElement() (boto.exception.SQSError method), 220
startElement() (boto.exception.StorageResponseError method), 220
startElement() (boto.gs.acl.ACL method), 540
startElement() (boto.gs.acl.Entries method), 540
startElement() (boto.gs.acl.Entry method), 540
startElement() (boto.gs.acl.Scope method), 540
startElement() (boto.gs.cors.Cors method), 559
startElement() (boto.gs.key.Key method), 572
startElement() (boto.gs.user.User method), 572
startElement() (boto.handler.XmlHandler method), 220
startElement() (boto.iam.summarymap.SummaryMap method), 588
startElement() (boto.mturk.connection.BaseAutoResultElement method), 600
startElement() (boto.mturk.price.Price method), 606
startElement() (boto.mws.response.ComplexAmount method), 623
startElement() (boto.mws.response.ComplexDimensions method), 623
startElement() (boto.mws.response.ComplexWeight method), 623
startElement() (boto.mws.response.Response method), 629
startElement() (boto.mws.response.ResponseElement method), 629
startElement() (boto.rds.dbinstance.DBInstance method), 650
startElement() (boto.rds.dbinstance.PendingModifiedValues method), 651
startElement() (boto.rds.dbinstance.ReadReplicaDBInstanceIdentifiers method), 651
startElement() (boto.rds.dbsecuritygroup.DBSecurityGroup method), 652
startElement() (boto.rds.dbsecuritygroup.EC2SecurityGroup method), 652
startElement() (boto.rds.dbsecuritygroup.IPRange method), 652
startElement() (boto.rds.dbsnapshot.DBSnapshot method), 653
startElement() (boto.rds.event.Event method), 653
startElement() (boto.rds.parametergroup.Parameter method), 654
startElement() (boto.rds.parametergroup.ParameterGroup method), 654
startElement() (boto.resultset.BooleanResult method), 220
startElement() (boto.resultset.ResultSet method), 221
startElement() (boto.route53.hostedzone.HostedZone method), 693

- startElement() (boto.route53.record.Record method), 693
- startElement() (boto.s3.acl.ACL method), 697
- startElement() (boto.s3.acl.Grant method), 698
- startElement() (boto.s3.acl.Policy method), 698
- startElement() (boto.s3.bucket.Bucket method), 709
- startElement() (boto.s3.cors.CORSConfiguration method), 715
- startElement() (boto.s3.cors.CORSRule method), 715
- startElement() (boto.s3.deletemarker.DeleteMarker method), 716
- startElement() (boto.s3.key.Key method), 726
- startElement() (boto.s3.lifecycle.Expiration method), 730
- startElement() (boto.s3.lifecycle.Lifecycle method), 731
- startElement() (boto.s3.lifecycle.Rule method), 731
- startElement() (boto.s3.lifecycle.Transitions method), 732
- startElement() (boto.s3.multidelete.Deleted method), 728
- startElement() (boto.s3.multidelete.Error method), 729
- startElement() (boto.s3.multidelete.MultiDeleteResult method), 729
- startElement() (boto.s3.multipart.CompleteMultiPartUpload method), 727
- startElement() (boto.s3.multipart.MultiPartUpload method), 728
- startElement() (boto.s3.multipart.Part method), 728
- startElement() (boto.s3.prefix.Prefix method), 727
- startElement() (boto.s3.tagging.Tag method), 732
- startElement() (boto.s3.tagging.Tags method), 732
- startElement() (boto.s3.tagging.TagSet method), 732
- startElement() (boto.s3.user.User method), 732
- startElement() (boto.sdb.domain.Domain method), 740
- startElement() (boto.sdb.domain.DomainDumpParser method), 741
- startElement() (boto.sdb.domain.DomainMetaData method), 741
- startElement() (boto.sdb.item.Item method), 742
- startElement() (boto.sqs.attributes.Attributes method), 755
- startElement() (boto.sqs.batchresults.BatchResults method), 769
- startElement() (boto.sqs.batchresults.ResultEntry method), 769
- startElement() (boto.sqs.message.RawMessage method), 763
- startElement() (boto.sqs.queue.Queue method), 768
- startElement() (boto.sts.credentials.AssumedRole method), 775
- startElement() (boto.sts.credentials.Credentials method), 776
- startElement() (boto.sts.credentials.DecodeAuthorizationMessage method), 776
- startElement() (boto.sts.credentials.FederationToken method), 776
- startElement() (boto.sts.credentials.User method), 776
- startElement() (boto.vpc.dhcpoptions.DhcpConfigSet method), 817
- startElement() (boto.vpc.dhcpoptions.DhcpOptions method), 817
- startElement() (boto.vpc.dhcpoptions.DhcpValueSet method), 817
- startElement() (boto.vpc.internetgateway.InternetGateway method), 818
- startElement() (boto.vpc.internetgateway.InternetGatewayAttachment method), 818
- startElement() (boto.vpc.routetable.Route method), 818
- startElement() (boto.vpc.routetable.RouteAssociation method), 818
- startElement() (boto.vpc.routetable.RouteTable method), 818
- startElement() (boto.vpc.vpnconnection.VpnConnection method), 820
- startElement() (boto.vpc.vpnconnection.VpnConnectionOptions method), 820
- startElement() (boto.vpc.vpnconnection.VpnStaticRoute method), 821
- startElement() (boto.vpc.vpnconnection.VpnTunnel method), 821
- startElement() (boto.vpc.vpngateway.Attachment method), 821
- startElement() (boto.vpc.vpngateway.VpnGateway method), 821
- Startup (class in boto.pyami.startup), 633
- state (boto.ec2.instance.Instance attribute), 447
- state_code (boto.ec2.instance.Instance attribute), 447
- Statistics (boto.ec2.cloudwatch.metric.Metric attribute), 292
- status (boto.dynamodb.table.Table attribute), 343
- status (boto.utils.ShellCommand attribute), 222
- Status (class in boto.ec2.instancestatus), 451
- Status (class in boto.route53.status), 695
- Step (class in boto.emr.emrobject), 515
- Step (class in boto.emr.step), 511
- StepConfig (class in boto.emr.emrobject), 515
- StepId (class in boto.emr.emrobject), 515
- StepSummary (class in boto.emr.emrobject), 515
- StepSummaryList (class in boto.emr.emrobject), 515
- stop() (boto.ec2.instance.Instance method), 447
- stop() (boto.pyami.installers.Installer method), 632
- stop() (boto.pyami.installers.ubuntu.installer.Installer method), 632
- stop() (boto.rds.dbinstance.DBInstance method), 650
- stop_all() (boto.ec2.instance.Reservation method), 449
- stop_configuration_recorder() (boto.configservice.layer1.ConfigServiceConnection method), 319
- stop_deployment() (boto.codedeploy.layer1.CodeDeployConnection method), 302
- stop_instance() (boto.opsworks.layer1.OpsWorksConnection

method), 867

stop_instances() (boto.ec2.connection.EC2Connection method), 438

stop_logging() (boto.cloudtrail.layer1.CloudTrailConnection method), 285

stop_stack() (boto.opsworks.layer1.OpsWorksConnection method), 867

stop_task() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 468

storage_class (boto.gs.key.Key attribute), 572

storage_class (boto.s3.key.Key attribute), 726

storage_class (boto.s3.lifecycle.Transitions attribute), 732

storage_uri() (in module boto), 213

storage_uri_for_key() (in module boto), 213

StorageClassBody (boto.gs.bucket.Bucket attribute), 541

StorageCopyError, 220

StorageCreateError, 220

StorageDataError, 220

StoragePermissionsError, 220

StorageQuotaExceeded, 876

StorageResponseError, 220

StreamingDistribution (class in boto.cloudfront.distribution), 238

StreamingDistributionConfig (class in boto.cloudfront.distribution), 239

StreamingDistributionSummary (class in boto.cloudfront.distribution), 239

StreamingObject (class in boto.cloudfront.object), 242

StreamingStep (class in boto.emr.step), 512

StringProperty (class in boto.sdb.db.property), 925

strip_namespace() (in module boto.mws.response), 630

STSConnection (class in boto.sts), 769

STYLE_XML_TEMPLATE (boto.mturk.question.SelectionAnswer attribute), 610

SubdomainCallingFormat (class in boto.s3.connection), 714

submit() (boto.dynamodb.batch.BatchList method), 346

submit() (boto.dynamodb.batch.BatchWriteList method), 347

submit_container_state_change() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 468

submit_feed() (boto.mws.connection.MWSConnection method), 620

submit_file() (boto.services.submit.Submitter method), 745

submit_path() (boto.services.submit.Submitter method), 745

submit_task_state_change() (boto.ec2containerservice.layer1.EC2ContainerServiceConnection method), 469

SubmitFeedResult (class in boto.mws.response), 629

Submitter (class in boto.services.submit), 745

Subnet (class in boto.vpc.subnet), 818

SubnetAlreadyInUse, 687, 876

SubParse (class in boto.ec2.instance), 449

subscribe() (boto.sns.SNSConnection method), 754

subscribe_sqs_queue() (boto.sns.SNSConnection method), 754

subscribe_to_dataset() (boto.cognito.sync.layer1.CognitoSyncConnection method), 315

Subscription (boto.mws.response.GetSubscriptionResult attribute), 626

Subscription (class in boto.mws.response), 630

SubscriptionAlreadyExist, 687, 876

SubscriptionCategoryNotFound, 687, 876

SubscriptionEventIdNotFound, 687

SubscriptionList (boto.mws.response.ListSubscriptionsResult attribute), 627

SubscriptionNotFound, 688, 876

SubscriptionRequiredException, 597

SubscriptionSeverityNotFound, 688

suggest() (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection method), 282

SummaryMap (class in boto.iam.summarymap), 588

SupportConnection (class in boto.support.layer1), 926

SupportedPermissions (in module boto.gs.acl), 540

SupportedProduct (class in boto.emr.emrobject), 515

suspend_processes() (boto.ec2.autoscale.AutoScaleConnection method), 181

suspend_processes() (boto.ec2.autoscale.group.AutoScalingGroup method), 183

SuspendedProcess (class in boto.ec2.autoscale.group), 184

swap_environment_cnames() (boto.beanstalk.layer1.Layer1 method), 199

SwapEnvironmentCNAMEsResponse (class in boto.beanstalk.response), 203

SwapEnvironmentCnamesResponse (class in boto.beanstalk.response), 203

SWFResponseError, 220

T

Table (class in boto.dynamodb.table), 338

Table (class in boto.dynamodb2.table), 357

table_from_schema() (boto.dynamodb.layer2.Layer2 method), 336

TableBatchGenerator (class in boto.dynamodb.table), 343

TableGenerator (class in boto.dynamodb.layer2), 337

Tag (class in boto.cloudformation.stack), 234

Tag (class in boto.ec2.autoscale.tag), 187

Tag (class in boto.ec2.tag), 460

Tag (class in boto.s3.tagging), 732

TaggedEC2Object (class in boto.ec2.ec2object), 440

Tags (class in boto.s3.tagging), 732

TagSet (class in boto.ec2.tag), 460

TagSet (class in boto.s3.tagging), 732

- TargetPrefix (boto.cloudhsm.layer1.CloudHSMConnection attribute), 243
- TargetPrefix (boto.cloudtrail.layer1.CloudTrailConnection attribute), 284
- TargetPrefix (boto.codedeploy.layer1.CodeDeployConnection attribute), 296
- TargetPrefix (boto.cognito.identity.layer1.CognitoIdentityConnection attribute), 307
- TargetPrefix (boto.configservice.layer1.ConfigServiceConnection attribute), 317
- TargetPrefix (boto.datapipeline.layer1.DataPipelineConnection attribute), 322
- TargetPrefix (boto.dynamodb2.layer1.DynamoDBConnection attribute), 370
- TargetPrefix (boto.kinesis.layer1.KinesisConnection attribute), 589
- TargetPrefix (boto.kms.layer1.KMSConnection attribute), 822
- TargetPrefix (boto.logs.layer1.CloudWatchLogsConnection attribute), 829
- TargetPrefix (boto.machinelearning.layer1.MachineLearningConnection attribute), 834
- TargetPrefix (boto.opsworks.layer1.OpsWorksConnection attribute), 847
- TargetPrefix (boto.route53.domains.layer1.Route53DomainsConnection attribute), 910
- TargetPrefix (boto.support.layer1.SupportConnection attribute), 927
- TaskNotFoundException, 327
- teardown() (boto.mws.response.DeclarativeType method), 623
- teardown() (boto.mws.response.MemberList method), 627
- template (boto.mturk.question.AnswerSpecification attribute), 607
- template (boto.mturk.question.Application attribute), 607
- template (boto.mturk.question.Binary attribute), 607
- template (boto.mturk.question.Constraints attribute), 608
- template (boto.mturk.question.ExternalQuestion attribute), 608
- template (boto.mturk.question.FileUploadAnswer attribute), 608
- template (boto.mturk.question.FormattedContent attribute), 608
- template (boto.mturk.question.FreeTextAnswer attribute), 608
- template (boto.mturk.question.HTMLQuestion attribute), 608
- template (boto.mturk.question.LengthConstraint attribute), 609
- template (boto.mturk.question.NumberOfLinesSuggestion attribute), 609
- template (boto.mturk.question.NumericConstraint attribute), 609
- template (boto.mturk.question.Overview attribute), 609
- template (boto.mturk.question.Question attribute), 609
- template (boto.mturk.question.QuestionContent attribute), 609
- template (boto.mturk.question.RegExConstraint attribute), 610
- template (boto.mturk.question.SimpleField attribute), 610
- Template (class in boto.cloudformation.template), 234
- TemplateParameter (class in boto.cloudformation.template), 234
- terminate() (boto.ec2.instance.Instance method), 448
- terminate() (boto.swf.layer2.WorkflowExecution method), 798
- terminate_environment() (boto.beanstalk.layer1.Layer1 method), 200
- terminate_instance() (boto.ec2.autoscale.AutoScaleConnection method), 181
- terminate_instances() (boto.ec2.connection.EC2Connection method), 438
- terminate_jobflow() (boto.emr.connection.EmrConnection method), 510
- terminate_jobflows() (boto.emr.connection.EmrConnection method), 510
- terminate_workflow_execution() (boto.swf.layer1.Layer1 method), 789
- TerminateEnvironmentResponse (class in boto.beanstalk.response), 204
- TerminationPolicies (class in boto.ec2.autoscale.group), 184
- TerminationPolicies (class in boto.ec2.autoscale.policy), 187
- test_metric_filter() (boto.logs.layer1.CloudWatchLogsConnection method), 832
- test_role() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491
- TextProperty (class in boto.sdb.db.property), 925
- throughput (boto.dynamodb2.fields.GlobalBaseIndexField attribute), 351
- ThruputError (boto.dynamodb.layer1.Layer1 attribute), 328
- TimeDecodeError, 921
- TimeProperty (class in boto.sdb.db.property), 925
- TLDRulesViolation, 918
- to_boolean() (boto.cloudfront.invalidation.InvalidationListResultSet method), 242
- to_boolean() (boto.resultset.BooleanResult method), 220
- to_boolean() (boto.resultset.ResultSet method), 221
- to_call() (boto.dynamodb2.results.ResultSet method), 357
- to_dict() (boto.dynamodb.batch.Batch method), 346
- to_dict() (boto.dynamodb.batch.BatchList method), 346
- to_dict() (boto.dynamodb.batch.BatchWrite method), 347
- to_dict() (boto.dynamodb.batch.BatchWriteList method), 347

- to_dict() (boto.sts.credentials.Credentials method), 776
- to_domain_connection_params()
 - (boto.cloudsearch2.search.Query method), 274
- to_json() (boto.cloudsearch.optionstatus.OptionStatus method), 257
- to_json() (boto.cloudsearch2.optionstatus.OptionStatus method), 273
- to_params() (boto.cloudsearch.search.Query method), 258
- to_params() (boto.cloudsearch2.search.Query method), 274
- to_print() (boto.route53.record.Record method), 693
- to_xml() (boto.cloudfront.distribution.DistributionConfig method), 238
- to_xml() (boto.cloudfront.distribution.StreamingDistributionConfig attribute), 709
- to_xml() (boto.cloudfront.distribution.StreamingDistributionConfig method), 239
- to_xml() (boto.cloudfront.identity.OriginAccessIdentityConfig method), 240
- to_xml() (boto.cloudfront.identity.OriginAccessIdentityConfig method), 240
- to_xml() (boto.cloudfront.invalidation.InvalidationBatch method), 241
- to_xml() (boto.cloudfront.origin.CustomOrigin method), 240
- to_xml() (boto.cloudfront.origin.S3Origin method), 240
- to_xml() (boto.ecs.item.ItemSet method), 470
- to_xml() (boto.ecs.item.ResponseGroup method), 471
- to_xml() (boto.gs.acl.ACL method), 540
- to_xml() (boto.gs.acl.Entries method), 540
- to_xml() (boto.gs.acl.Entry method), 540
- to_xml() (boto.gs.acl.Scope method), 540
- to_xml() (boto.gs.cors.Cors method), 559
- to_xml() (boto.gs.user.User method), 572
- to_xml() (boto.route53.healthcheck.HealthCheck method), 692
- to_xml() (boto.route53.record.Record method), 693
- to_xml() (boto.route53.record.ResourceRecordSets method), 695
- to_xml() (boto.s3.acl.ACL method), 697
- to_xml() (boto.s3.acl.Grant method), 698
- to_xml() (boto.s3.acl.Policy method), 698
- to_xml() (boto.s3.cors.CORSConfiguration method), 715
- to_xml() (boto.s3.cors.CORSRule method), 715
- to_xml() (boto.s3.lifecycle.Expiration method), 730
- to_xml() (boto.s3.lifecycle.Lifecycle method), 731
- to_xml() (boto.s3.lifecycle.Rule method), 731
- to_xml() (boto.s3.lifecycle.Transition method), 731
- to_xml() (boto.s3.lifecycle.Transitions method), 732
- to_xml() (boto.s3.multipart.MultiPartUpload method), 728
- to_xml() (boto.s3.tagging.Tag method), 732
- to_xml() (boto.s3.tagging.Tags method), 732
- to_xml() (boto.s3.tagging.TagSet method), 732
- to_xml() (boto.s3.user.User method), 732
- to_xml() (boto.sdb.db.query.Query method), 926
- to_xml() (boto.sdb.domain.Domain method), 740
- TooManyRecordsException, 220
- TooManyRequestsException, 311, 316
- Trac (class in boto.pyami.installers.ubuntu.trac), 633
- track_progress_less_iterations()
 - (boto.gs.resumable_upload_handler.ResumableUploadHandler method), 573
- TrackingEvents (boto.mws.response.GetPackageTrackingDetailsResult attribute), 625
- TradeInValue (boto.mws.response.CompetitivePricing attribute), 623
- TrailAlreadyExistsException, 288
- TrailNotFoundException, 288
- TrailNotProvidedException, 288
- trans_region (boto.s3.bucket.S3WebsiteEndpointTranslate method), 709
- transfer_domain() (boto.route53.domains.layer1.Route53DomainsConnection method), 914
- TransferThread (class in boto.glacier.concurrent), 539
- Transition (class in boto.s3.lifecycle), 731
- Transitions (class in boto.s3.lifecycle), 731
- translate_region() (boto.s3.bucket.S3WebsiteEndpointTranslate class method), 709
- TreeHashDoesNotMatchError, 539
- Trigger (class in boto.beanstalk.response), 204
- trim_snapshots() (boto.ec2.connection.EC2Connection method), 439
- TrustedSigners (class in boto.cloudfront.signers), 241
- type_name (boto.sdb.db.property.BlobProperty attribute), 921
- type_name (boto.sdb.db.property.BooleanProperty attribute), 922
- type_name (boto.sdb.db.property.DateProperty attribute), 922
- type_name (boto.sdb.db.property.DateTimeProperty attribute), 922
- type_name (boto.sdb.db.property.FloatProperty attribute), 922
- type_name (boto.sdb.db.property.IntegerProperty attribute), 923
- type_name (boto.sdb.db.property.ListProperty attribute), 923
- type_name (boto.sdb.db.property.LongProperty attribute), 923
- type_name (boto.sdb.db.property.MapProperty attribute), 923
- type_name (boto.sdb.db.property.PasswordProperty attribute), 924
- type_name (boto.sdb.db.property.Property attribute), 924
- type_name (boto.sdb.db.property.ReferenceProperty attribute), 925
- type_name (boto.sdb.db.property.S3KeyProperty attribute), 925
- type_name (boto.sdb.db.property.StringProperty attribute), 925

- tribute), 925
 - type_name (boto.sdb.db.property.TextProperty attribute), 925
 - type_name (boto.sdb.db.property.TimeProperty attribute), 925
- ## U
- umount() (boto.pyami.scriptbase.ScriptBase method), 633
 - unassign_instance() (boto.opsworks.layer1.OpsWorksConnection method), 867
 - unassign_private_ip_addresses() (boto.ec2.connection.EC2Connection method), 439
 - unassign_volume() (boto.opsworks.layer1.OpsWorksConnection method), 867
 - UnauthorizedOperation, 688
 - UnauthorizedOperationFault, 688
 - unblock_worker() (boto.mturk.connection.MTurkConnection method), 604
 - UnexpectedHTTPResponseError, 539
 - UnfulfillablePreviewItems (boto.mws.response.FulfillmentPreview attribute), 624
 - Units (boto.ec2.cloudwatch.metric.Metric attribute), 292
 - UnknownFilterTypeError, 402
 - UnknownIndexFieldError, 402
 - UnknownSchemaFieldError, 402
 - UnknownSnapshotCopyRegion, 688
 - unlink_developer_identity() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 310
 - unlink_identity() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 310
 - unmonitor() (boto.ec2.instance.Instance method), 448
 - unmonitor_instance() (boto.ec2.connection.EC2Connection method), 439
 - unmonitor_instances() (boto.ec2.connection.EC2Connection method), 439
 - unquote_v() (in module boto.utils), 224
 - unshare() (boto.ec2.snapshot.Snapshot method), 458
 - unsubscribe() (boto.sns.SNSConnection method), 754
 - unsubscribe_from_dataset() (boto.cognito.sync.layer1.CognitoSyncConnection method), 315
 - UnsupportedOperationException, 828
 - UnsupportedOption, 688
 - UnsupportedOptionFault, 688
 - UnsupportedTLD, 918
 - update() (boto.cloudformation.stack.Stack method), 233
 - update() (boto.cloudfront.distribution.Distribution method), 237
 - update() (boto.cloudfront.distribution.StreamingDistribution method), 238
 - update() (boto.cloudfront.identity.OriginAccessIdentity method), 240
 - update() (boto.dynamodb2.table.Table method), 368
 - update() (boto.ec2.autoscale.group.AutoScalingGroup method), 183
 - update() (boto.ec2.cloudwatch.alarm.MetricAlarm method), 295
 - update() (boto.ec2.elb.healthcheck.HealthCheck method), 501
 - update() (boto.ec2.image.Image method), 444
 - update() (boto.ec2.instance.Instance method), 448
 - update() (boto.ec2.networkinterface.NetworkInterface method), 453
 - update() (boto.ec2.snapshot.Snapshot method), 458
 - update() (boto.ec2.volume.Volume method), 462
 - update() (boto.rds.dbinstance.DBInstance method), 650
 - update() (boto.rds.dbsnapshot.DBSnapshot method), 653
 - update() (boto.route53.status.Status method), 695
 - update() (boto.sqs.message.MHMessage method), 763
 - update() (boto.vpc.vpc.VPC method), 820
 - update_a() (boto.route53.zone.Zone method), 696
 - update_access_key() (boto.iam.connection.IAMConnection method), 585
 - update_account_password_policy() (boto.iam.connection.IAMConnection method), 585
 - update_alarm() (boto.ec2.cloudwatch.CloudWatchConnection method), 292
 - update_app() (boto.opsworks.layer1.OpsWorksConnection method), 867
 - update_application() (boto.beanstalk.layer1.Layer1 method), 200
 - update_application() (boto.codedeploy.layer1.CodeDeployConnection method), 302
 - update_application_version() (boto.beanstalk.layer1.Layer1 method), 200
 - update_assume_role_policy() (boto.iam.connection.IAMConnection method), 586
 - update_availability_options() (boto.cloudsearch2.layer1.CloudSearchConnection method), 271
 - update_batch_prediction() (boto.machinelearning.layer1.MachineLearningConnection method), 845
 - update_classic_link_enabled() (boto.vpc.vpc.VPC method), 820
 - update_cname() (boto.route53.zone.Zone method), 697
 - update_configuration_template() (boto.beanstalk.layer1.Layer1 method), 201
 - update_data_source() (boto.machinelearning.layer1.MachineLearningConnection method), 845
 - update_default_search_field() (boto.cloudsearch.layer1.Layer1 method),

255

update_deployment_group() (boto.codedeploy.layer1.CodeDeployConnection method), 302

update_dme() (in module boto.utils), 224

update_domain_contact() (boto.route53.domains.layer1.Route53DomainsConnection method), 916

update_domain_contact_privacy() (boto.route53.domains.layer1.Route53DomainsConnection method), 916

update_domain_nameservers() (boto.route53.domains.layer1.Route53DomainsConnection method), 917

update_elastic_ip() (boto.opsworks.layer1.OpsWorksConnection method), 868

update_environment() (boto.beanstalk.layer1.Layer1 method), 201

update_evaluation() (boto.machinelearning.layer1.MachineLearningConnection method), 845

update_from_data() (boto.cloudsearch.domain.Domain method), 250

update_from_data() (boto.cloudsearch2.domain.Domain method), 266

update_from_response() (boto.dynamodb.table.Table method), 343

update_function_configuration() (boto.awslambda.layer1.AWSLambdaConnection method), 190

update_global_secondary_index() (boto.dynamodb2.table.Table method), 368

update_group() (boto.iam.connection.IAMConnection method), 586

update_identity_pool() (boto.cognito.identity.layer1.CognitoIdentityConnection method), 310

update_inbound_shipment() (boto.mws.connection.MWSConnection method), 621

update_instance() (boto.opsworks.layer1.OpsWorksConnection method), 868

update_item() (boto.dynamodb.layer1.Layer1 method), 331

update_item() (boto.dynamodb.layer2.Layer2 method), 336

update_item() (boto.dynamodb2.layer1.DynamoDBConnection method), 394

update_key_description() (boto.kms.layer1.KMSConnection method), 827

update_layer() (boto.opsworks.layer1.OpsWorksConnection method), 869

update_login_profile() (boto.iam.connection.IAMConnection method), 587

update_metadata() (boto.gs.key.Key method), 572

update_metadata() (boto.s3.key.Key method), 726

update_ml_model() (boto.machinelearning.layer1.MachineLearningConnection method), 845

update_mx() (boto.route53.zone.Zone method), 697

update_my_user_profile() (boto.opsworks.layer1.OpsWorksConnection method), 870

update_pipeline() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 491

update_pipeline_notifications() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 494

update_pipeline_status() (boto.elastictranscoder.layer1.ElasticTranscoderConnection method), 494

update_qualification_score() (boto.mturk.connection.MTurkConnection method), 604

update_qualification_type() (boto.mturk.connection.MTurkConnection method), 604

update_rds_db_instance() (boto.opsworks.layer1.OpsWorksConnection method), 871

update_record() (boto.route53.zone.Zone method), 697

update_records() (boto.cognito.sync.layer1.CognitoSyncConnection method), 315

update_report_acknowledgements() (boto.mws.connection.MWSConnection method), 621

update_saml_provider() (boto.iam.connection.IAMConnection method), 587

update_scaling_parameters() (boto.cloudsearch2.layer1.CloudSearchConnection method), 271

update_server_cert() (boto.iam.connection.IAMConnection method), 587

update_service_access_policies() (boto.cloudsearch.layer1.Layer1 method), 255

update_service_access_policies() (boto.cloudsearch2.layer1.CloudSearchConnection method), 271

update_signing_cert() (boto.iam.connection.IAMConnection method), 587

update_size() (boto.cloudsearch.search.Query method), 258

update_size() (boto.cloudsearch2.search.Query method), 274

update_stack() (boto.cloudformation.connection.CloudFormationConnection method), 230

update_stack() (boto.opsworks.layer1.OpsWorksConnection method), 871

update_stemming_options() (boto.cloudsearch.layer1.Layer1 method),

255

update_stopword_options() (boto.cloudsearch.layer1.Layer1 method), 255

update_subscription() (boto.mws.connection.MWSConnection method), 621

update_synonym_options() (boto.cloudsearch.layer1.Layer1 method), 256

update_table() (boto.dynamodb.layer1.Layer1 method), 331

update_table() (boto.dynamodb2.layer1.DynamoDBConnection method), 401

update_throughput() (boto.dynamodb.layer2.Layer2 method), 337

update_throughput() (boto.dynamodb.table.Table method), 343

update_trail() (boto.cloudtrail.layer1.CloudTrailConnection method), 286

update_user() (boto.iam.connection.IAMConnection method), 587

update_user_profile() (boto.opsworks.layer1.OpsWorksConnection method), 873

update_volume() (boto.opsworks.layer1.OpsWorksConnection method), 873

UpdateApplicationResponse (class in boto.beanstalk.response), 204

UpdateApplicationVersionResponse (class in boto.beanstalk.response), 204

UpdateConfigurationTemplateResponse (class in boto.beanstalk.response), 204

UpdateEnvironmentResponse (class in boto.beanstalk.response), 204

UpdateReportAcknowledgementsResult (class in boto.mws.response), 630

upload() (boto.glacier.concurrent.ConcurrentUploader method), 539

upload_archive() (boto.glacier.layer1.Layer1 method), 531

upload_archive() (boto.glacier.vault.Vault method), 536

upload_documents() (boto.cloudsearchdomain.layer1.CloudSearchDomainConnection method), 283

upload_function() (boto.awslambda.layer1.AWSLambdaConnection method), 191

upload_id (boto.glacier.writer.Writer attribute), 537

upload_part() (boto.glacier.layer1.Layer1 method), 532

upload_part_from_file() (boto.s3.multipart.MultiPartUpload method), 728

upload_server_cert() (boto.iam.connection.IAMConnection method), 588

upload_signing_cert() (boto.iam.connection.IAMConnection method), 588

UploadArchiveError, 540

UploaderThread (class in boto.sdb.domain), 741

UploadWorkerThread (class in boto.glacier.concurrent), 539

uri() (boto.cloudfront.identity.OriginAccessIdentity method), 240

url() (boto.cloudfront.object.Object method), 242

url() (boto.cloudfront.object.StreamingObject method), 242

Usage (boto.services.bs.BS attribute), 743

usage() (in module boto.pyami.launch_ami), 633

use_boolean() (boto.dynamodb2.table.Table method), 369

use_decimals() (boto.dynamodb.layer2.Layer2 method), 337

use_ip() (boto.ec2.instance.Instance method), 448

User (class in boto.gs.user), 572

User (class in boto.s3.user), 732

User (class in boto.sts.credentials), 776

USWest (boto.s3.connection.Location attribute), 711

USWest2 (boto.s3.connection.Location attribute), 711

V

validate_request_intervals (boto.route53.healthcheck.HealthCheck attribute), 692

valid_states (boto.cloudformation.connection.CloudFormationConnection attribute), 232

valid_states (boto.cloudformation.stack.StackEvent attribute), 233

ValidApplyMethods (boto.rds.parametergroup.Parameter attribute), 653

ValidApplyTypes (boto.rds.parametergroup.Parameter attribute), 653

validate() (boto.mturk.question.ValidatingXML method), 611

validate() (boto.sdb.db.property.DateProperty method), 922

validate() (boto.sdb.db.property.DateTimeProperty method), 922

validate() (boto.sdb.db.property.FloatProperty method), 922

validate() (boto.sdb.db.property.IntegerProperty method), 923

validate() (boto.sdb.db.property.ListProperty method), 923

validate() (boto.sdb.db.property.LongProperty method), 923

validate() (boto.sdb.db.property.MapProperty method), 923

validate() (boto.sdb.db.property.PasswordProperty method), 924

validate() (boto.sdb.db.property.Property method), 924

validate() (boto.sdb.db.property.ReferenceProperty method), 925

validate() (boto.sdb.db.property.S3KeyProperty method), 925

validate() (boto.sdb.db.property.TextProperty method), 925
 validate() (boto.sdb.db.property.TimeProperty method), 925
 validate_configuration_settings() (boto.beanstalk.layer1.Layer1 method), 202
 validate_get_all_versions_params() (boto.gs.bucket.Bucket method), 555
 validate_get_all_versions_params() (boto.s3.bucket.Bucket method), 709
 validate_kwarg_names() (boto.gs.bucket.Bucket method), 555
 validate_kwarg_names() (boto.s3.bucket.Bucket method), 709
 validate_pipeline_definition() (boto.datapipeline.layer1.DataPipelineConnection method), 326
 validate_regex (boto.sdb.db.property.S3KeyProperty attribute), 925
 validate_string() (in module boto.sdb.db.property), 925
 validate_template() (boto.cloudformation.connection.CloudFormationConnection method), 232
 ValidateConfigurationSettingsResponse (class in boto.beanstalk.response), 204
 validateParseLevel() (boto.gs.cors.Cors method), 559
 ValidatingXML (class in boto.mturk.question), 611
 ValidationError (boto.dynamodb.layer1.Layer1 attribute), 328
 ValidationException, 321, 402, 495, 874
 ValidationMessage (class in boto.beanstalk.response), 204
 ValidSources (boto.rds.parametergroup.Parameter attribute), 653
 ValidTypes (boto.rds.parametergroup.Parameter attribute), 653
 ValidValues (boto.ec2.instance.InstanceAttribute attribute), 448
 value (boto.rds.parametergroup.Parameter attribute), 654
 values() (boto.dynamodb2.items.Item method), 356
 values() (boto.sqs.message.MHMessage method), 763
 values() (boto.utils.LazyLoadMetadata method), 222
 VariationRelationship (class in boto.mws.response), 630
 vault (boto.glacier.writer.Writer attribute), 537
 Vault (class in boto.glacier.vault), 533
 verbose_name (boto.sdb.db.property.Property attribute), 924
 verify() (boto.mturk.notification.NotificationMessage method), 606
 verify_domain_dkim() (boto.ses.connection.SESConnection method), 748
 verify_domain_identity() (boto.ses.connection.SESConnection method), 748
 verify_email_address() (boto.ses.connection.SESConnection method), 748
 verify_email_identity() (boto.ses.connection.SESConnection method), 749
 verify_signature() (boto.fps.connection.FPSConnection method), 520
 Version (boto.cloudfront.CloudFrontConnection attribute), 234
 Version (boto.dynamodb.layer1.Layer1 attribute), 328
 Version (boto.glacier.layer1.Layer1 attribute), 521
 Version (boto.route53.connection.Route53Connection attribute), 688
 versioned_bucket_list() (in module boto.gs.bucketlistresultset), 555
 versioned_bucket_list() (in module boto.s3.bucketlistresultset), 710
 VersionedBucketListResultSet (class in boto.gs.bucketlistresultset), 555
 VersionedBucketListResultSet (class in boto.s3.bucketlistresultset), 710
 VersioningBody (boto.s3.bucket.Bucket attribute), 698
 VersioningConfiguration (boto.s3.bucket.Bucket attribute), 698
 VHostCallingFormat (class in boto.s3.connection), 714
 Volume (class in boto.ec2.volume), 460
 volume_state() (boto.ec2.volume.Volume method), 462
 VolumeAttribute (class in boto.ec2.volume), 462
 VolumeStatus (class in boto.ec2.volumestatus), 463
 VolumeStatusSet (class in boto.ec2.volumestatus), 463
 VPC (class in boto.vpc.vpc), 818
 VPCCConnection (class in boto.vpc), 801
 VpnConnection (class in boto.vpc.vpnconnection), 820
 VpnConnectionOptions (class in boto.vpc.vpnconnection), 820
 VpnGateway (class in boto.vpc.vpngateway), 821
 VpnStaticRoute (class in boto.vpc.vpnconnection), 820
 VpnTunnel (class in boto.vpc.vpnconnection), 821

W

WAIT_BEFORE_RETRY (boto.exception.ResumableTransferDisposition attribute), 219
 wait_for_state() (boto.cloudsearch.optionstatus.OptionStatus method), 257
 WorkflowExecution (class in boto.swf.layer2), 797
 workflows() (boto.swf.layer2.Domain method), 796
 WorkflowType (class in boto.swf.layer2), 799
 write() (boto.glacier.writer.Writer method), 537
 write() (boto.sqs.queue.Queue method), 768
 write_batch() (boto.sqs.queue.Queue method), 768
 write_message() (boto.services.service.Service method), 744
 write_message() (boto.services.submit.Submitter method), 745
 write_metadata() (boto.pyami.bootstrap.Bootstrap method), 630

`write_mime_multipart()` (in module `boto.utils`), 224
`write_off_debt()` (`boto.fps.connection.FPSConnection`
method), 520
`write_units` (`boto.dynamodb.table.Table` attribute), 343
`Writer` (class in `boto.glacier.writer`), 537
`WRRBody` (`boto.route53.record.Record` attribute), 693

X

`xml_template` (`boto.mturk.question.QuestionForm`
attribute), 610
`XMLBody` (`boto.route53.record.Record` attribute), 693
`XMLFQDNPart` (`boto.route53.healthcheck.HealthCheck`
attribute), 692
`XmlHandler` (class in `boto.handler`), 220
`XmlHandlerWrapper` (class in `boto.handler`), 220
`XMLIpAddrPart` (`boto.route53.healthcheck.HealthCheck`
attribute), 692
`XMLNameSpace` (`boto.route53.connection.Route53Connection`
attribute), 688
`XMLRequestIntervalPart`
(`boto.route53.healthcheck.HealthCheck` at-
tribute), 692
`XMLStringMatchPart` (`boto.route53.healthcheck.HealthCheck`
attribute), 692
`XMLTemplate` (class in `boto.mturk.question`), 611

Y

`YAMLMessage` (class in `boto.contrib.ymlmessage`), 321

Z

`Zone` (class in `boto.ec2.zone`), 463
`Zone` (class in `boto.route53.zone`), 695