

---

# **botoform Documentation**

*Release 0.0.1*

**Russell Ballestrini**

**Jun 23, 2017**



---

## Contents

---

<b>1 Quickstart</b>	<b>3</b>
1.1 Quickstart . . . . .	3
<b>2 Schema Reference</b>	<b>7</b>
2.1 Schema Reference . . . . .	7
2.2 Hello World . . . . .	10
<b>3 Tools</b>	<b>11</b>
3.1 Tools . . . . .	11
<b>4 Developer Reference</b>	<b>15</b>
4.1 Developer Reference . . . . .	15
<b>Python Module Index</b>	<b>35</b>



### Manage infrastructure running on AWS using YAML templates.

*Botoform* provides tools to manage the lifecycle of related AWS resources. We use a simple YAML schema to document resources as infrastructure. The YAML schema has self documenting qualities and works with version control.

In this example we use the *bf create* tool to build the infrastructure defined in *helloworld.yaml*:

The *bf* tools use YAML architecture to create and manage environments. *Botoform* allows reproduction of any environment, no matter how complex.

*Botoform* abstracts and enriches the *Boto3* and *Botocore* projects.



## Quickstart

### Contents

- *Quickstart*
  - *Installation*
    - \* *Virtualenv*
    - \* *Automatic install from source with botoform-bootstrap.sh*
    - \* *Manual install from source*
    - \* *Verify the botoform install*
  - *Configuration*
  - *Using Botoform*
    - \* *Create VPC*
    - \* *Unlock VPC*
    - \* *Destroy VPC*
  - *Example Output*

## Installation

### Virtualenv

Both the automatic and manual install assume that the `virtualenv` tool is installed. If you do not have `virtualenv` installed, you may do the following:

```
# only run this if you are missing the virtualenv tool.
sudo pip install virtualenv
```

### Automatic install from source with botoform-bootstrap.sh

---

**Note:** You should always review scripts prior to piping them from the Internet into your shell.

---

This script automates the steps in the *manual install from source* section.

The following one-liner will install botoform (bf) into your home directory:

```
wget -O - https://raw.githubusercontent.com/russellballestrini/botoform/master/
↳botoform-bootstrap.sh | sh
```

Next you should *verify the botoform install*.

### Manual install from source

Clone botoform repo:

```
git clone https://github.com/russellballestrini/botoform.git $HOME/botoform
cd $HOME/botoform
```

Create and activate a Python virtualenv named env:

```
virtualenv env
. env/bin/activate
```

Install dependencies into virtualenv:

```
python setup.py develop
```

Next you should *verify the botoform install*.

### Verify the botoform install

Whenever you want to use the bf tool, you need to activate the virtualenv:

You may verify installation by running:

```
bf --help
```

You should see usage information.

Next, edit your AWS *configuration* file with your access/secret keys.

## Configuration

Setup your AWS CLI config file, for example -

```
~/.aws/config:
```

```
[profile development]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
region = us-west-2
```

You are now ready to begin *using Botoform!*

## Using Botoform

### Create VPC

**Note:** This section will create real resources on AWS.

```
bf --profile=development --region=ap-southeast-1 create dogtest01 -e 'vpc_cidr=192.
↪168.1.0/24' tests/fixtures/webapp.yaml
```

### Unlock VPC

**Note:** This command will unlock instances to allow them to be terminated.

Disable API Termination Protection on all instances in VPC.

```
bf --profile=development --region=ap-southeast-1 unlock dogtest01
```

### Destroy VPC

**Danger:** This command will completely destroy the entire VPC and all related resources!

```
bf --profile=development --region=ap-southeast-1 destroy dogtest01
```

## Example Output

```
bf -p development -r us-east-1 create -e 'vpc_cidr=10.20.20.0/24' stg example.yaml

creating vpc (stg, 10.20.20.0/24)
tagging vpc (Name:stg)
modifying vpc for dns support
modifying vpc for dns hostnames
creating internet_gateway (igw-stg)
tagging gateway (Name:igw-stg)
attaching igw to vpc (igw-stg)
creating DHCP Options Set for stg
associating DHCP Options dopt-30731454 with VPC stg
creating route_table (stg-public)
tagging route_table (Name:stg-public)
```

```
creating route_table (stg-private)
tagging route_table (Name:stg-private)
creating subnet 10.20.20.96/27 in us-east-1c
tagging subnet (Name:stg-private-2)
creating subnet 10.20.20.64/27 in us-east-1b
tagging subnet (Name:stg-private-1)
creating subnet 10.20.20.32/27 in us-east-1b
tagging subnet (Name:stg-public-1)
creating subnet 10.20.20.0/27 in us-east-1c
tagging subnet (Name:stg-public-2)
creating security_group stg-all
tagging security_group (Name:stg-all)
creating security_group stg-bastion
tagging security_group (Name:stg-bastion)
creating key pair default
associating rt private with sn private-2
associating rt private with sn private-1
associating rt public with sn public-1
associating rt public with sn public-2
creating role: bastion
1 instances of role bastion launching into stg-public-1 subnet
inbound rule: 'bastion' -> 'all' over ports 22 (TCP)
inbound rule: '98.110.147.178/32' -> 'bastion' over ports 22 (TCP)
inbound rule: '184.188.101.86/32' -> 'bastion' over ports 22 (TCP)
waiting for i-00b858ff5644d8634 to start
waiting for i-00b858ff5644d8634 to be in status OK
tagging instance i-00b858ff5644d8634 (Name:stg-bastion-delawareriver)
tagging volumes for instance stg-bastion-delawareriver (Name:stg-bastion-
↳delawareriver)
allocating eip and associating with stg-bastion-delawareriver
allocated eip 54.80.219.185 and associated with stg-bastion-delawareriver
locking new normal (not autoscaled) instances to prevent termination
adding route ['0.0.0.0/0', 'internet_gateway'] to route_table (stg-public)
managing route53 private zone.
done! don't you look awesome. : )
```

End user YAML Schema Reference for describing VPCs and related AWS resources.

### Schema Reference

End user YAML Schema Reference for describing VPCs and related AWS resources.

These documents describe how to structure YAML in a Botoform compatible schema.

You may optionally use Jinja2 in your YAML config.

#### amis

Define AMIs using the following schema:

```
amis:
  human-readable-custom-name:
    region-name: ami-id
```

Example code snippet from `amis.yaml`:

```
amis:
  ubuntu-12.04-lts:
    us-east-1: ami-59a4a230
    us-west-1: ami-660c3023
    us-west-2: ami-fa9cf1ca
    sa-east-1: ami-1da90a00
    eu-west-1: ami-808675f7
    eu-central-1: ami-043d0b19
```

```
ap-southeast-1: ami-3c39686e
ap-southeast-2: ami-09f26b33
ap-northeast-1: ami-f381f5f2
```

```
ubuntu-12.04-lts-hvm:
```

```
us-east-1: ami-5fa4a236
us-west-1: ami-620c3027
us-west-2 : ami-fc9cf1cc
sa-east-1: ami-1fa90a02
```

## route\_tables

Define `route_tables` using the following schema:

```
route_tables:
  human-readable-custom-name:
    routes:
      - ['cidr', 'destination']
```

Example code snippet from `route_tables.yaml`:

```
# This is an example route_tables schema.
route_tables:

  private-1:
    routes:
      - ['0.0.0.0/0', 'nat']

  private-2:
    routes:
      - ['0.0.0.0/0', 'nat']

  private-3:
    routes:
      - ['0.0.0.0/0', 'nat']

  public:
    routes:
      - ['0.0.0.0/0', 'internet_gateway']
```

## subnets

Define subnets using the following schema:

```
subnets:

  human-readable-custom-name:

    size: cidr_integer
    route_table: name-of-the-route-table
    description: human readable description (optional)
    availability_zone: char of az (optional)
```

Example code snippet from `subnets.yaml`:

```
# This is an example of a 4 subnet schema.
# Each subnet has a size of 27 which is a /27 CIDR (32 addresses).
# Two are attached to the private route_table, and two to the public.
# We round robin AZ letters if availability_zone is not defined.
# Instances will launch into subnet with public IPs, if public is True.
subnets:

  private-1:
    size: 27
    route_table: private

  private-2:
    size: 27
    route_table: private

  public-1:
    size: 27
    route_table: public
    availability_zone: a

  public-2:
    size: 27
    route_table: public
    availability_zone: b
```

## tags

Optional tags to add to all taggable resources using the following schema:

```
tags:
  key: value
```

For example to create the `env` tag with the value `dev` add this to your schema:

```
tags:
  env: dev
```

## vpc

Optional vpc schema settings:

```
vpc_cidr: the-cidr-block-to-allocate-to-your-vpc
vpc_tenancy: whether-or-not-your-vpc-should-use-default-or-dedicated
```

For example:

```
vpc_cidr: 10.10.10.0/24
vpc_tenancy: default
```

## Hello World

Smallest botoform template, helloworld.yaml:

```
vpc_cidr: {{ vpc_cidr }}

amis:
  ubuntu-14.04-lts-hvm:
    us-east-1: ami-fce3c696

route_tables:
  public:
    routes:
      - ['0.0.0.0/0', 'internet_gateway']

subnets:
  public-1:
    size: 27
    route_table: public
    public: True

security_groups:
  bastion:
    inbound:
      - ['0.0.0.0/0', 'tcp', 22]

instance_roles:
  bastion:
    instance_type: t2.micro
    ami: 'ubuntu-14.04-lts-hvm'
    count: 1
    security_groups: ['bastion']
    subnets: ['public-1']
    eip: true
    block_devices:
      "/dev/sda1":
        size: 10
```

You may optionally use Jinja2 in your YAML config.

## Tools

### bf

Botoform tools are namespaced under the `bf` command.

For a list built-in subcommands and interactive help, run `bf --help`.

Currently Implemented subcommands:

```
{atmosphere, shell, cli, dump, list, lock, create, stop, start, unlock, repl, destroy}
```

### list

List all existing VPC names.

For example:

```
bf list
```

You can also pass the particular AWS profile and/or region:

```
bf --profile development --region us-west-2 list
```

### create

Create a new VPC and related services, modeled after the given YAML template.

For example:

```
bf create dogtest01 -e 'vpc_cidr=192.168.1.0/24' tests/fixtures/webapp.yaml
```

## **destroy**

**Warning:** this is destructive!

Destroy a VPC and all related services.

For example:

```
bf destroy dogtest01
```

## **refresh**

Refresh VPC by adding resources defined but missing in given YAML template.

So far we have implemented the following subcommands:

- `instance_roles`
- `security_groups`
- `load_balancers`
- `tags`
- `private_zone`

## **reflect**

---

**Note:** not implemented yet.

---

**Warning:** this is destructive!

Make VPC reflect given YAML template by adding and removing resources.

- `instance_roles`
- `security_groups`
- `tags`
- `private_zone`

## **stop**

Stop all instances in VPC including autoscaled instances.

**Warning:** Currently does `_NOT_` skip “ephemeral” instances!

### start

Start all instances in VPC including autoscaled instances.

### lock

Enable API Termination Protection on all instances in VPC.

### unlock

Disable API Termination Protection on all instances in VPC.

### repl

Open an interactive REPL (read-eval-print-loop) with access to evpc object.

Once you have a repl, try running *evpc.roles* or *evpc.instances*.

```
usage: bf repl vpc_name [-h]
```

**Note:** Install *bpython* into your environment for more fun.

```
bf webapp01 repl

You now have access to the evpc object, for example: evpc.roles

>>> evpc.instances
[<botoform.enriched.instance.EnrichedInstance object at 0x10e194350>,
<botoform.enriched.instance.EnrichedInstance object at 0x10e1944d0>]

>>> map(str, evpc.instances)
['webapp01-web01', 'webapp01-web02']
```

### cli

An alias to *repl* so it works the same.

### shell

An alias to *repl* so it works the same.

### dump

Output existing resources or services in a Botoform compatible format.

- instances
- security\_groups
- ansible\_hosts
- tags

## **atmosphere**

For every AWS profile + region, dump every VPC to STDOUT.

This command takes a while to run, so you should likely redirect the output to a file.

Reason for this tool is we have many AWS accounts and we use many regions.

Using the output of this tool, we can easily grep for a vpc\_name and find where it lives.

Developer Python Library Reference for modifying or extending Botoform.

## Developer Reference

Developer Reference for modifying or extending Botoform.

This document provides details for Botoform's Python codebase.

### builders.py

The builders.py file uses YAML config to build the environment.

```
class botoform.builders.EnvironmentBuilder(vpc_name, config=None, region_name=None,  
                                           profile_name=None, log=None)
```

```
    add_eip_to_instance(instance)
```

```
    apply_all()  
        Build the environment specified in the config.
```

```
    associate_route_tables_with_subnets(subnet_cfg)
```

```
    attach_vpn_gateway(vpn_gateway_cfg)  
        Attach defined VPN gateway to VPC
```

```
    autoscaling_instance_role(role_name, role_data, desired_count)
```

```
    autoscaling_instance_roles(instance_role_cfg)  
        Create Autoscaling Groups and Launch Configurations.
```

```
    build_internet_gateway()  
        Build and attach Internet Gateway to VPC.
```

**build\_vpc** (*cidrblock='172.31.0.0/16', tenancy='default'*)  
Build VPC

**create\_dhcp\_options** (*dhcp\_configurations*)  
Creates and return a new dhcp\_options set.

**create\_instance\_profile** (*instance\_profile\_name*)  
Create instance\_profile and role, return instance\_profile.

**db\_instances** (*db\_instance\_cfg*)  
Build RDS DB Instances.

**dhcp\_options** (*dhcp\_options\_cfg*)  
Creates DHCP Options Set and associates with VPC

**endpoints** (*route\_tables*)  
Build VPC endpoints for given route\_tables

**finish\_instance\_roles** (*instance\_role\_cfg, instances=None*)

**get\_instance\_profile** (*instance\_profile\_name*)  
Return instance\_profile or None.

**instance\_profiles** (*instance\_role\_cfg*)

**instance\_role** (*role\_name, role\_data, desired\_count*)

**instance\_roles** (*instance\_role\_cfg*)  
Create instance roles defined in config.

**key\_pairs** (*key\_pair\_cfg*)

**load\_balancers** (*load\_balancer\_cfg*)  
Build ELB load balancers.

**route\_table\_rules** (*route\_cfg*)  
Build route table rules defined in config

**route\_tables** (*route\_cfg*)  
Build route\_tables defined in config

**security\_group\_inbound\_rules** (*security\_group\_cfg*)  
Build inbound rule for Security Group defined in config.

**security\_group\_outbound\_revoke\_default\_rule** (*sg*)

**security\_group\_outbound\_rules** (*security\_group\_cfg*)  
Build outbound rule for Security Group defined in config.

**security\_group\_rule\_to\_permission** (*rule*)  
Return a permission dictionary from a rule tuple.

**security\_group\_rules** (*security\_group\_cfg*)  
Build Security Group Rules defined in config.

**security\_group\_rules\_to\_permissions** (*sg\_name, rules, direction='inbound'*)

**security\_groups** (*security\_group\_cfg*)  
Build Security Groups defined in config.

**subnets** (*subnet\_cfg*)  
Build subnets defined in config.

**tag\_instance\_name** (*\*args, \*\*kw*)  
Accept a EnrichedInstance, objects create tags.

**tag\_instance\_volumes** (*instance*)  
Accept an EnrichedInstance, tag all attached volumes.

**tags** (*tag\_cfg*)

**wait\_for\_instance\_profile** (*\*args, \*\*kw*)

**wait\_for\_instance\_roles\_to\_exist** (*\*args, \*\*kw*)

`botoform.builders.get_default_ec2_trust_policy` (*region\_name*)

## config.py

The config.py turns YAML into Python dictionaries, supports Jinja2.

**class** `botoform.config.ConfigLoader` (*template\_dir=None, context\_vars=None*)  
Loads *Botoform Schema* and returns a dictionary.

**load** (*template\_path=None, template\_string=None*)  
Load a *Botoform Schema* config and render with Jinja2.

### Parameters

- **template\_path** – Path to the config to load and render.
- **template\_string** – String to load and render. Optional.

**Returns** Python dictionary representation of *config*.

**render** (*template\_file*)  
Return Jinja2 render of template with context\_vars.

**render\_string** (*template\_string*)  
Return Jinja2 render of template\_str with context\_vars.

**template\_dir**

## subnetallocator.py

`botoform.subnetallocator.allocate` (*cidrs, sizes*)  
Accept network block CIDRs and list of subnetwork CIDR sizes. Return a list of IPNetwork objects allocated from the network block.

For example:

```
>>> allocate('10.10.10.0/24', [28,27,29])
[IPNetwork('10.10.10.0/27'), IPNetwork('10.10.10.32/28'), IPNetwork('10.10.10.48/
↪29')]
```

## util.py

The util.py file is sort of a junk drawer right now... but..

---

**class** `botoform.util.BotoConnections` (*region\_name=None, profile\_name=None*)  
Central Management of boto3 client and resource connection objects.

**azones**

Return a list of available AZ names for active AWS profile/region.

**profile\_name**

**refresh\_boto\_connections** ()

Attach related Boto3 clients and resources.

**region\_name**

**setup\_session\_and\_refresh\_connections** ()

```
class botoform.util.BotoformDumper (stream, default_style=None, default_flow_style=None,
canonical=None, indent=None, width=None, allow_unicode=None, line_break=None, encoding=None,
explicit_start=None, explicit_end=None, version=None, tags=None)
```

A custom YAML dumper that is pretty.

**increase\_indent** (flow=False, indentless=False)

**yaml\_representers** = {<type 'float': <unbound method SafeRepresenter.represent\_float>, <type 'set': <unbound m

```
class botoform.util.Log (desired_level='debug', syslog=True, stdout=True, program='botoform')
```

Handles emitting logs to syslog and stdout.

**emit** (message, level='info')

Emit message if level meets requirement.

**message:** Any object that has a `__str__` method.

**level:** The level or severity of this message (default info)

**levels**

Return a list of levels on and beyond desired\_level.

```
botoform.util.collection_len (collection)
```

```
botoform.util.collection_to_list (collection)
```

```
botoform.util.dict_to_key_value (data, sep='=', pair_sep=',')
```

Return a string representation of a dictionary.

by default this function will turn:

```
{'key1': 'value1', 'key2': 'value2'}
```

into:

```
key1=value1, key2=value2
```

#### Parameters

- **data** – The dictionary to convert into a string.
- **sep** – Optional, string to separate keys and values (Default '=')
- **pair\_sep** – Optional, string to separate key/value pairs (Default ',')

**Returns** a string representation of the given dictionary.

```
botoform.util.generate_password (size=9, pool=None)
```

Return a system generated password.

#### Parameters

- **size** – The desired length of the password to generate. (Default 9)
- **pool** – Pool of chars to choose from. (Default digits and letters [upper/lower])

**Returns** String (raw password)

`botoform.util.get_block_device_map_from_role_config` (*role\_cfg*)  
accept role config data and return a Boto3 friendly BlockDeviceMappings.

`botoform.util.get_ids` (*objects*)  
Return a list of ids from a list of objects.

**Parameters** *objects* – A list of objects all of whom have an id attribute.

**Returns** A list of ids

`botoform.util.get_port_range` (*raw\_range*, *ip\_protocol='tcp'*)  
Returns a (from\_port, to\_port) tuple.

Examples:

```
>>> get_port_range(443)
(443, 443)

>>> get_port_range('all')
(1, 65535)

>>> get_port_range('5000-5009')
(5000, 5009)

>>> get_port_range(' 8080')
(8080, 8080)

>>> get_port_range('tacobell', ip_protocol='icmp')
(-1, -1)
```

#### Parameters

- **raw\_range** – A string or integer.
- **ip\_protocol** – Optional, 'tcp', 'udp', 'icpm' (Default 'tcp')

**Returns** (from\_port, to\_port)

`botoform.util.id_to_human` (*id\_string*)  
Turn an id into a human readable hash digest.

**Parameters** *id\_string* – The subject string to generate a human hash of.

```
>>> id_to_human('i-ceebb70c')
'friendisland'
```

`botoform.util.key_value_to_dict` (*key\_value\_list*, *sep='='*, *pair\_sep=', '*)  
Return a dictionary from a list of key/value strings.

turns *key\_value\_list*, like:

```
key_value_list = ['a=1,b=2', 'c=3, d=4', 'e=5']
```

into a dict, like:

```
{'a':'1', 'b':'2', 'c':'3', 'd':'4', 'e':'5'}
```

**Parameters**

- **key\_value\_list** – The list of key/value strings to convert into a dict.
- **sep** – Optional, string which separates keys and values (Default ‘=’)
- **pair\_sep** – Optional, string which separates key/value pairs (Default ‘,’)

**Returns** A string representation of the given dictionary.

`botoform.util.make_filter` (*key, values*)

Return a filter document expected by Boto3.

**Parameters**

- **key** – The key name for this new filter document.
- **values** – A value or a list of values to filter/match on.

**Returns** A filter document (list/dict) in the form that Boto3 expects.

`botoform.util.make_tag_dict` (*ec2\_object*)

Return a dictionary of existing tags.

**Parameters** *ec2\_object* – A tagable Boto3 object with a tags attribute.

**Returns** A dictionary where tag names are keys and tag values are values.

`botoform.util.map_filter_false` (*function, items*)

Works like map but automatically filters out untruthy items.

```
>>> map_filter_false(lambda i : i, [1,2,None,3,False,0,4])  
[1, 2, 3, 4]
```

**Parameters**

- **function** – the function to map items through
- **items** – a list of items to map through the function

**Returns** list

`botoform.util.merge_pages` (*key, pages*)

Merge boto3 paginated results into single list.

**Parameters**

- **key** – The document key to merge from all pages.
- **pages** – An iterator of page documents.

**Returns** A single flat list containing results of all pages.

`botoform.util.normalize_sg_port` (*sg\_rule\_tuple*)

accept a security group rule tuple, return normalized port range.

`botoform.util.normalize_sg_rules` (*sg\_rules*)

accept a list of security group rule tuples, return list with normalized port ranges.

`botoform.util.output_formatter` (*data, output\_format='newline'*)

Print data in the optional output\_format.

`botoform.util.reflect_attrs` (*child, parent, skip\_attrs=None*)  
Composition Magic: reflect all missing parents attributes into child.

**Parameters**

- **child** – Object to receive attributes.
- **parent** – Object to source attributes from.
- **skip\_attrs** – Optional list of attrs strings to not reflect.

**Returns** None

`botoform.util.snake_to_camel_case` (*name, answers=None*)  
Accept a snake\_case string and return a CamelCase string.

For example:

```
>>> snake_to_camel_case('cidr_block')
'CidrBlock'
```

`botoform.util.tag_filter` (*tag\_key, tag\_values*)  
Return a tag filter document expected by boto3.

**Parameters**

- **tag\_key** – A tag key or name.
- **tag\_values** – The tag value or a list of values to filter on.

**Returns** A filter document (list/dict) in the form that Boto3 expects.

`botoform.util.update_tags` (*\*args, \*\*kw*)  
Add or update tags to reflect given keyword args

**Parameters**

- **ec2\_object** – A tagable Boto3 object with a tags attribute.
- **\*\*kwargs** – key=value where key is tag name, value is tag value.

**Returns** None

`botoform.util.write_private_key` (*key\_pair*)  
Write private key to filesystem.

**Parameters** **key\_pair** – The Boto3 KeyPair object to write to filesystem.

**Returns** None

## enriched

Botoform abstracts and enriches the [Boto3](#) and [Botocore](#) projects.

The references in this document describe how we extend each project.

We use composition to provide many helper attributes and methods.

### autoscaling.py

`class botoform.enriched.autoscaling.EnrichedAutoscaling` (*evpc*)

```
    delete_related_autoscaling_groups ()
```

```
delete_related_launch_configs ()
get_all_autoscaling_group_descriptions ()
    Return a list of all autoscaling groups descriptions.
get_all_launch_config_descriptions ()
    Return a list of all launch configuration descriptions.
get_related_autoscaling_group_descriptions ()
    Return a list of related autoscaling groups descriptions.
get_related_autoscaling_group_names ()
get_related_launch_config_descriptions ()
    Return a list of related launch configuration descriptions.
get_related_launch_config_names ()
scale_related_autoscaling_group (autoscaling_group_name, count)
```

## elasticache.py

```
class botoform.enriched.elasticache.EnrichedElasticCache (evpc)
```

```
delete_related_cache_clusters (cluster_ids=None)
    delete all cache clusters and subnet groups related to this VPC.
    Parameters cluster_ids – optional list of cache_cluster_ids (names) to delete instead.
    Returns None
get_all_cluster_descriptions ()
get_all_subnet_group_descriptions ()
get_related_cluster_descriptions ()
    return a list of cache cluster descriptions related to VPC.
get_related_cluster_endpoints ()
    return a list of cache cluster dns endpoints related to this VPC
get_related_cluster_ids ()
    return a list of cache cluster ids related to this VPC
get_related_subnet_group_descriptions ()
    return a list of cache subnet group descriptions related to VPC.
wait_for_related_clusters (waiter_name, cluster_ids=None)
    wait for related dbs to transition to desired state.
```

## elb.py

```
class botoform.enriched.elb.EnrichedElb (evpc)
```

```
delete_related_elbs ()
format_instance_ids (instance_ids)
    god boto3 is a pain sometimes.
format_listeners (listener_tuples)
```

**get\_all\_elb\_descriptions** ()  
Return a list of all ELB (Load Balancer) descriptions.

**get\_related\_elb\_descriptions** ()  
Return a list of related ELB (Load Balancer) descriptions.

**get\_related\_elb\_names** ()  
Return a list of related ELB (Load Balancer) names.

**register\_role\_with\_load\_balancer** (*elb\_name, role\_name*)

## instance.py

**class** botoform.enriched.instance.**EnrichedInstance** (*instance, evpc=None*)

This class uses composition to enrich Boto3's Instance resource class.

ec2.Instance(boto3.resources.base.ServiceResource)

Reference:

<https://boto3.readthedocs.org/en/latest/reference/services/ec2.html#instance>

**allocate\_and\_associate\_eip** ()  
Allocate and Associate a new EIP with instance and return EIP.

**Returns** New VpcAddress EIP object

**allocate\_eip** ()  
Allocate a new EIP and return allocation\_id

**Returns** New allocation\_id

**associate\_eip** (*\*args, \*\*kw*)  
Associate EIP with instance and return EIP.

**Returns** New VpcAddress EIP object

**autoscale\_group**  
Return autoscaling group name (AWS aws:autoscaling:groupName tag).

**disassociate\_eips** (*release=True*)  
Disassociate all EIPs associated with this instance.

**Parameters** **release** – Also release allocations for EIPs. Default True.

**Return type** None

**eips**  
Return a list of VpcAddress objects associated to this instance. :rtype: list

**hostname**  
Return hostname (AWS Name tag).

**id\_human**  
Return humanhash of id.

**identifiers**  
Return a tuple of “unique” identifier strings for instance. :rtype: tuple

**identity**  
Return hostname (AWS Name tag) or id.

**is\_autoscaled**  
Return True if this instance was autoscaled else False

**lock()**  
Lock this instance to prevent termination.

**name**  
Return hostname (AWS Name tag).

**reflect\_attrs()**  
reflect all attributes of ec2.Instance into self.

**reload()**  
run the reload method on the attached instance and reflect\_attrs.

**role**  
Return role from from 'role' or 'Name' tag: web, db, ...

**shortname**  
Return shortname from hostname: web-solarmaine, db-beerindia, ...

**source\_dest\_check\_disable()**  
Disable source destination checking. Needed for NATs and Routers.

**source\_dest\_check\_enable()**  
Enable source destination checking. Default.

**tag\_dict**  
Return dictionary of tags.

**unlock()**  
Unlock this instance to allow termination.

**wait\_until\_status\_ok()**  
Wait (block) until this instance state is 'OK'.

## key\_pair.py

```
class botoform.enriched.key_pair.EnrichedKeyPair (evpc)
```

```
    create_key_pair (short_key_pair_name)  
        Create a KeyPair object, update key_pairs AWS tag.
```

```
    delete_key_pair (short_key_pair_name)  
        Delete a KeyPair object, update key_pairs AWS tag.
```

```
    delete_key_pairs ()  
        Delete ALL related KeyPair objects, update key_pairs AWS tag.
```

```
    delete_key_pairs_tag ()
```

```
    get_key_name (short_key_pair_name)  
        Returns full key_name if key exists, else None
```

```
    get_key_pair (short_key_pair_name)  
        Return a KeyPair object by key_name.
```

```
    key_names  
        Return a list of ssh key pair names from AWS VPC tag.
```

```
    key_pairs  
        Return a dictionary of ssh KeyPair objects.
```

## rds.py

`class botoform.enriched.rds.EnrichedRds (evpc)`

`delete_related_db_instances (db_ids=None, skip_snapshot=False)`

delete all RDS DB instances and subnet groups related to this VPC.

`rds_ids`: optional list of rds\_ids (names) to delete instead.

`get_all_db_descriptions ()`

return a list of all db description dictionaries.

`get_related_connection_data ()`

return a dictionary of DB Connection data related to this VPC

`get_related_db_descriptions ()`

return a list of db description dictionaries related to this VPC.

`get_related_db_endpoints ()`

return a list of cache cluster dns endpoints related to this VPC

`get_related_db_ids ()`

return a list of db instance identifiers related to this VPC.

`reset_master_passwords (db_ids)`

Iterate over list of db\_ids (names) reset master password immediately. Return dictionary of changes in the form of {'db\_id': 'new\_pass'}.

`wait_for_related_dbs (waiter_name, db_ids=None)`

wait for related dbs to transition to desired state.

## route53.py

`class botoform.enriched.route53.EnrichedRoute53 (evpc)`

`change_private_zone (change_docs)`

`create_private_zone ()`

`delete_private_zone ()`

`empty_private_zone ()`

`list_all_private_resource_record_sets ()`

`list_all_resource_record_sets (hosted_zone_id, record_sets=None, marker=None)`

Handles pagination unlike list\_resource\_record\_sets.

`private_zone_id`

get the vpc with the private hosted zone id from vpc tag.

`private_zone_name`

`refresh_private_zone ()`

## vpc.py

`class botoform.enriched.vpc.EnrichedVPC (vpc_name=None, region_name=None, pro-  
file_name=None, log=None)`

This class uses composition to enrich Boto3's VPC resource class. Here we relate AWS resources using var-

ious techniques like the `vpc_name` tag. We also provide methods for managing the lifecycle of related AWS resources.

**associate\_route\_table\_with\_subnet** (*rt\_name, sn\_name*)

Accept a route table name and subnet name, associate them.

**attach\_vpn\_gateway** (*vgw\_id*)

Attach VPN gateway to the VPC

**azones**

**connect** (*vpc\_name*)

connect to VPC and reflect all attributes into self.

**delete\_dhcp\_options** ()

Delete DHCP Options Set

**delete\_instances** (*instances=None, wait=True*)

Terminate all or a list of instances.

**delete\_internet\_gateways** ()

Delete related internet gateways.

**delete\_route\_tables** ()

Delete related route tables.

**delete\_security\_group** (*\*args, \*\*kw*)

**delete\_security\_groups** ()

Delete related security groups.

**delete\_subnets** ()

Delete related subnets.

**detach\_vpn\_gateway** ()

Detach VPN gateway from VPC

**enriched\_security\_groups**

Format Security Groups (and permissions) in *Botoform Schema*.

**Returns** security\_groups in *Botoform Schema*.

**ensure\_vgw\_state** (*\*args, \*\*kw*)

if vgw is not in expected state, throw exception.

**exclude\_instances** (*identifiers=None, roles=None*)

Accept a list of identifiers and/or roles. Return a list of instances which do *not* match either qualifier list.

**Note:** This method returns all instances if both identifiers and roles is None.

**find\_instance** (*identifier*)

Return an instance or None which matches identifier.

Raises exception if multiple instances match identifier.

**Parameters identifier** –

A list of identifiers to qualify instances by, for example:

- custid-ui01
- ui01
- 192.168.1.9
- i-01234567

**Returns** EnrichedInstance or None

**find\_instances** (*identifiers=None, roles=None, exclude=False*)

Accept a list of identifiers and/or roles. Return a list of instances which match either qualifier list.

**Parameters**

- **identifiers** –

**Optional, a list of identifiers to qualify instances by, for example:**

- custid-ui01
- ui01
- 192.168.1.9
- i-01234567

- **roles** –

**Optional, a list of roles to qualify instances by, for example:**

- ui
- api
- proxy

- **exclude** – If True, qualifiers exclude instead of include! Defaults to False.

**Danger:** This method will return *no* instances if all qualifiers are None. However, if *exclude* is True we could return *all* instances!

**Returns** A list of EnrichedInstance objets or an empty list.

**get\_autoscaled\_instances** (*instances=None*)

return a list of instances which were created via autoscaling.

**get\_instances** (*instances=None*)

Returns a possibly empty list of EnrichedInstance objects.

**Parameters** **instances** – Optional, list or collection to convert to EnrichedInstance objects.

**Returns** list of EnrichedInstance objects

**get\_main\_route\_table** ()

Return the main (default) route table for VPC.

**get\_normal\_instances** (*instances=None*)

return a list of instances which were *\_not\_* created via autoscaling.

**get\_role** (*role\_name, instances=None*)

Return a possibly empty list of EnrichedInstance objects.

**Parameters**

- **role\_name** – The name of the role whose instances to return.
- **instances** – Optional, list or collection to search role from.

**Returns** A list of EnrichedInstance objects.

**get\_roles** (*instances=None*)

Return a dict of lists where role is the key and a list of EnrichedInstance objects is the value.

**get\_route\_table** (*name*)  
Accept route table name, return route\_table object or None.

**get\_running\_instances** (*instances=None*)  
Return list running EnrichedInstance object related to this VPC.

**get\_security\_group** (*name*)  
Accept security group name, return security group object or None.

**get\_subnet** (*name*)  
Accept subnet name, return subnet object or None.

**get\_vgw** (*vgw\_id*)  
Accept vgw\_id and return vgw description.

**get\_vpc\_by\_name\_tag** (*vpc\_name*)  
lookup vpc by vpc\_name tag. Raises exceptions on insanity.

**get\_vpn\_gateways** ()  
Gets all the VGWs attached to the VPC

**identity**

**include\_instances** (*identifiers=None, roles=None*)  
Accept a list of identifiers and/or roles. Return a list of instances which match either qualifier list.  
  
**Note:** This method returns no instances if both identifiers and roles is None.

**instances**

**lock\_instances** (*instances=None*)  
Lock all or a list of instances.

**name**

**reflect\_attrs** ()  
reflect all attributes of boto3's vpc resource object into self.

**region\_name**

**reload** ()  
run the reload method on the attached instance and reflect\_attrs.

**revoke\_inbound\_rules\_from\_sg** (*sg*)

**revoke\_outbound\_rules\_from\_sg** (*sg*)

**revoke\_security\_group\_rules** (*sg*)

**roles**

**start\_instances** (*instances=None, wait=True*)  
Start all or a list of instances.

**stop\_instances** (*instances=None, wait=True*)  
Stop all or a list of instances.

**tag\_dict**

**taggable\_resources**  
Return a list of taggable objects related to this VPC.

**terminate** ()  
Terminate all resources related to this VPC!

**unlock\_instances** (*instances=None*)

Unlock all or a list of instances.

**wait\_until\_instances** (*instances=None, state=None*)

## vpc\_endpoint.py

**class** botoform.enriched.vpc\_endpoint.**EnrichedVpcEndpoint** (*evpc*)

**create\_all** (*route\_table\_names*)

Accept route table names, create all endpoints for route tables.

**delete\_related** ()

Delete all VPC endpoints related to this VPC.

**describe\_related** ()

Return VPC endpoint descriptions related to this VPC.

**related\_ids** ()

Return VPC endpoint ids related to this VPC.

**services** ()

Return a list of available VPC endpoint services.

## plugins

You may extend the *bf* tool by writing a plugin.

A *bf* plugin must take one of two forms:

### function plugin

This example shows how to write a function plugin for destroying VPCs.

First we define an *entry point* and subcommand named `destroy`.

*setup.py*:

```
entry_points = {
    'botoform.plugins' : [
        'destroy = mybotoform.plugins.destroy:destroy',
    ]
}
```

The *entry point* named `destroy` shows the path to the `destroy` function.

*bf* function plugins must accept an *args* object and an *evpc* object.

For Example:

*mybotoform/plugins/destroy.py*:

```
def destroy(args, evpc):
    """
    Destroy a VPC and related resources and services.

    :param args: The parsed arguments and flags from the CLI.
    :param evpc: An instance of :meth:`botoform.enriched.vpc.EnrichedVPC`.
```

```
:returns: None
"""
evpc.terminate()
```

## class plugin

This example shows how to write a class plugin to dump instances to STDOUT.

We show how to structure your code to define additional subparser arguments.

---

**Note:** Use a *function plugin* instead of a class plugin if your subcommand does not need to define additional flags or args.

---

*setup.py:*

```
entry_points = {
    'botoform.plugins' : [
        'dump-instances = mybotoform.plugins.dump:Instances',
    ]
}
```

*mybotoform/plugins/dump.py:*

```
from botoform.util import output_formatter

class Instances(object):
    """Output a list of instance names. (example botoform plugin)"""

    @staticmethod
    def setup_parser(parser):
        parser.add_argument('--output-format',
            choices=['csv', 'yaml', 'json', 'newline'], default='newline',
            help='the desired format of any possible output')

    @staticmethod
    def main(args, evpc):
        """Output a list of instance names. (example botoform plugin)"""
        instances = evpc.instances
        print(output_formatter(map(str, instances), args.output_format))
```

This is a special class that has two staticmethods.

**setup\_parser:** accepts subcommand parser and allows additional flags and args to be defined.

**main:** main logic for this plugin.

Regardless of the form of plugin you choose, your plugin project's *setup.py* must define an *entry point* in the `botoform.plugins` group. The name of the *entry point* will be the subcommand on the CLI.

All *bf* subcommands (*core plugins*) are implemented in this way.

**See also:**

working examples [plugins](#) directory.

## core plugins

### core plugins

The Botoform *bf* tool ships with many core plugins and subcommands.

#### botoform.plugins.create

**class** `botoform.plugins.create.Create`

Create a new VPC and related services, modeled from YAML template.

This is a *class plugin* for the *bf* tool.

**static main** (*args*, *evpc=None*)

Creates a new VPC and related services, modeled from a YAML template.

##### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – `botoform.enriched.vpc.EnrichedVPC()` or `None`.

**Returns** `None`

**static setup\_parser** (*parser*)

Accepts a subparser and attaches additional arguments and flags.

**Parameters** **parser** – An ArgumentParser sub parser. Reference: <https://docs.python.org/3/library/argparse.html>

**Returns** `None`

#### botoform.plugins.destroy

`botoform.plugins.destroy.destroy` (*args*, *evpc*)

Destroy a VPC and related resources and services.

##### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** `None`

#### botoform.plugins.lock

`botoform.plugins.lock.lock` (*args*, *evpc*)

Lock all instances in VPC to prevent termination.

##### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** `None`

## botoform.plugins.unlock

`botoform.plugins.unlock.unlock` (*args*, *evpc*)

Unock all instances in VPC to allow termination.

### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** None

## botoform.plugins.dump

`class botoform.plugins.dump.Dump`

Dump AWS resourses as *Botoform Schema*.

This is a *class plugin* for the *bf* tool.

**static main** (*args*, *evpc*)

**static setup\_parser** (*parser*)

`botoform.plugins.dump.ansible_hosts` (*args*, *evpc*)

Output an Ansible host inventory for the VPC.

### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** Ansible inventory to standard out.

`botoform.plugins.dump.instances` (*args*, *evpc*)

Output instance roles to standard out in *Botoform Schema*.

### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** instance\_roles to standard out in *Botoform Schema*.

`botoform.plugins.dump.security_groups` (*args*, *evpc*)

Output Security Groups to standard out in *Botoform Schema*.

### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Returns** security\_groups to standard out in *Botoform Schema*.

## botoform.plugins.repl

`botoform.plugins.repl.REPL` (*args*, *evpc*)

Open an interactive REPL (read-eval-print-loop) with access to evpc object

### Parameters

- **args** – The parsed arguments and flags from the CLI.
- **evpc** – An instance of `botoform.enriched.vpc.EnrichedVPC()`.

**Subcommand aliases** cli and shell

**Returns** Interactive shell with evpc `botoform.enriched.vpc.EnrichedVPC()`.

## terms

### entry point

**entry point:** An entry point is a Python object identified in a project's `setup.py` file. The object is referenced by group and name to make it discoverable.

This means that another Python application can search for installed software. During the search, often the entry point group filters relevant objects.

Botoform uses this method to allow plugins to load at run time.



**b**

`botoform.builders`, 15  
`botoform.config`, 17  
`botoform.enriched.autoscaling`, 21  
`botoform.enriched.elasticache`, 22  
`botoform.enriched.elb`, 22  
`botoform.enriched.instance`, 23  
`botoform.enriched.key_pair`, 24  
`botoform.enriched.rds`, 25  
`botoform.enriched.route53`, 25  
`botoform.enriched.vpc`, 25  
`botoform.enriched.vpc_endpoint`, 29  
`botoform.plugins.create`, 31  
`botoform.plugins.destroy`, 31  
`botoform.plugins.dump`, 32  
`botoform.plugins.lock`, 31  
`botoform.plugins.repl`, 32  
`botoform.plugins.unlock`, 32  
`botoform.subnetallocator`, 17  
`botoform.util`, 17



## A

add\_eip\_to\_instance() (botoform.builders.EnvironmentBuilder method), 15  
 allocate() (in module botoform.subnetallocator), 17  
 allocate\_and\_associate\_eip() (botoform.enriched.instance.EnrichedInstance method), 23  
 allocate\_eip() (botoform.enriched.instance.EnrichedInstance method), 23  
 ansible\_hosts() (in module botoform.plugins.dump), 32  
 apply\_all() (botoform.builders.EnvironmentBuilder method), 15  
 associate\_eip() (botoform.enriched.instance.EnrichedInstance method), 23  
 associate\_route\_table\_with\_subnet() (botoform.enriched.vpc.EnrichedVPC method), 26  
 associate\_route\_tables\_with\_subnets() (botoform.builders.EnvironmentBuilder method), 15  
 attach\_vpn\_gateway() (botoform.builders.EnvironmentBuilder method), 15  
 attach\_vpn\_gateway() (botoform.enriched.vpc.EnrichedVPC method), 26  
 autoscale\_group (botoform.enriched.instance.EnrichedInstance attribute), 23  
 autoscaling\_instance\_role() (botoform.builders.EnvironmentBuilder method), 15  
 autoscaling\_instance\_roles() (botoform.builders.EnvironmentBuilder method), 15  
 azones (botoform.enriched.vpc.EnrichedVPC attribute), 26  
 azones (botoform.util.BotoConnections attribute), 17

## B

BotoConnections (class in botoform.util), 17  
 botoform.builders (module), 15  
 botoform.config (module), 17  
 botoform.enriched.autoscaling (module), 21  
 botoform.enriched.elasticache (module), 22  
 botoform.enriched.elb (module), 22  
 botoform.enriched.instance (module), 23  
 botoform.enriched.key\_pair (module), 24  
 botoform.enriched.rds (module), 25  
 botoform.enriched.route53 (module), 25  
 botoform.enriched.vpc (module), 25  
 botoform.enriched.vpc\_endpoint (module), 29  
 botoform.plugins.create (module), 31  
 botoform.plugins.destroy (module), 31  
 botoform.plugins.dump (module), 32  
 botoform.plugins.lock (module), 31  
 botoform.plugins.repl (module), 32  
 botoform.plugins.unlock (module), 32  
 botoform.subnetallocator (module), 17  
 botoform.util (module), 17  
 BotoformDumper (class in botoform.util), 18  
 build\_internet\_gateway() (botoform.builders.EnvironmentBuilder method), 15  
 build\_vpc() (botoform.builders.EnvironmentBuilder method), 15

## C

change\_private\_zone() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 collection\_len() (in module botoform.util), 18  
 collection\_to\_list() (in module botoform.util), 18  
 ConfigLoader (class in botoform.config), 17  
 connect() (botoform.enriched.vpc.EnrichedVPC method), 26  
 Create (class in botoform.plugins.create), 31

create\_all() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 21  
     method), 29  
 create\_dhcp\_options() (botoform.builders.EnvironmentBuilder method), 16  
 create\_instance\_profile() (botoform.builders.EnvironmentBuilder method), 16  
 create\_key\_pair() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24  
 create\_private\_zone() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 delete\_route\_tables() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_security\_group() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_security\_groups() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_subnets() (botoform.enriched.vpc.EnrichedVPC method), 26  
 describe\_related() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 29  
 destroy() (in module botoform.plugins.destroy), 31  
 detach\_vpn\_gateway() (botoform.enriched.vpc.EnrichedVPC method), 26  
 dhcp\_options() (botoform.builders.EnvironmentBuilder method), 16  
 dict\_to\_key\_value() (in module botoform.util), 18  
 disassociate\_eips() (botoform.enriched.instance.EnrichedInstance method), 23  
 Dump (class in botoform.plugins.dump), 32  
**D**  
 db\_instances() (botoform.builders.EnvironmentBuilder method), 16  
 delete\_dhcp\_options() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_instances() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_internet\_gateways() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_key\_pair() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24  
 delete\_key\_pairs() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24  
 delete\_key\_pairs\_tag() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24  
 delete\_private\_zone() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 delete\_related() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 29  
 delete\_related\_autoscaling\_groups() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 21  
 delete\_related\_cache\_clusters() (botoform.enriched.elasticache.EnrichedElastiCache method), 22  
 delete\_related\_db\_instances() (botoform.enriched.rds.EnrichedRds method), 25  
 delete\_related\_elbs() (botoform.enriched.elb.EnrichedElb method), 22  
 delete\_related\_launch\_configs() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 21  
 delete\_security\_group() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_security\_groups() (botoform.enriched.vpc.EnrichedVPC method), 26  
 delete\_subnets() (botoform.enriched.vpc.EnrichedVPC method), 26  
 describe\_related() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 29  
 destroy() (in module botoform.plugins.destroy), 31  
 detach\_vpn\_gateway() (botoform.enriched.vpc.EnrichedVPC method), 26  
 dhcp\_options() (botoform.builders.EnvironmentBuilder method), 16  
 dict\_to\_key\_value() (in module botoform.util), 18  
 disassociate\_eips() (botoform.enriched.instance.EnrichedInstance method), 23  
 Dump (class in botoform.plugins.dump), 32  
**E**  
 eips (botoform.enriched.instance.EnrichedInstance attribute), 23  
 emit() (botoform.util.Log method), 18  
 empty\_private\_zone() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 endpoints() (botoform.builders.EnvironmentBuilder method), 16  
 enriched\_security\_groups (botoform.enriched.vpc.EnrichedVPC attribute), 26  
 EnrichedAutoscaling (class in botoform.enriched.autoscaling), 21  
 EnrichedElastiCache (class in botoform.enriched.elasticache), 22  
 EnrichedElb (class in botoform.enriched.elb), 22  
 EnrichedInstance (class in botoform.enriched.instance), 23  
 EnrichedKeyPair (class in botoform.enriched.key\_pair), 24  
 EnrichedRds (class in botoform.enriched.rds), 25  
 EnrichedRoute53 (class in botoform.enriched.route53), 25  
 EnrichedVPC (class in botoform.enriched.vpc), 25  
 EnrichedVpcEndpoint (class in botoform.enriched.vpc\_endpoint), 29

ensure\_vgw\_state() (botoform.enriched.vpc.EnrichedVPC method), 26

EnvironmentBuilder (class in botoform.builders), 15

exclude\_instances() (botoform.enriched.vpc.EnrichedVPC method), 26

## F

find\_instance() (botoform.enriched.vpc.EnrichedVPC method), 26

find\_instances() (botoform.enriched.vpc.EnrichedVPC method), 27

finish\_instance\_roles() (botoform.builders.EnvironmentBuilder method), 16

format\_instance\_ids() (botoform.enriched.elb.EnrichedElb method), 22

format\_listeners() (botoform.enriched.elb.EnrichedElb method), 22

## G

generate\_password() (in module botoform.util), 18

get\_all\_autoscaling\_group\_descriptions() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_all\_cluster\_descriptions() (botoform.enriched.elasticache.EnrichedElastiCache method), 22

get\_all\_db\_descriptions() (botoform.enriched.rds.EnrichedRds method), 25

get\_all\_elb\_descriptions() (botoform.enriched.elb.EnrichedElb method), 22

get\_all\_launch\_config\_descriptions() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_all\_subnet\_group\_descriptions() (botoform.enriched.elasticache.EnrichedElastiCache method), 22

get\_autoscaled\_instances() (botoform.enriched.vpc.EnrichedVPC method), 27

get\_block\_device\_map\_from\_role\_config() (in module botoform.util), 19

get\_default\_ec2\_trust\_policy() (in module botoform.builders), 17

get\_ids() (in module botoform.util), 19

get\_instance\_profile() (botoform.builders.EnvironmentBuilder method), 16

get\_instances() (botoform.enriched.vpc.EnrichedVPC method), 27

get\_key\_name() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24

get\_key\_pair() (botoform.enriched.key\_pair.EnrichedKeyPair method), 24

get\_main\_route\_table() (botoform.enriched.vpc.EnrichedVPC method), 27

get\_normal\_instances() (botoform.enriched.vpc.EnrichedVPC method), 27

get\_port\_range() (in module botoform.util), 19

get\_related\_autoscaling\_group\_descriptions() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_related\_autoscaling\_group\_names() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_related\_cluster\_descriptions() (botoform.enriched.elasticache.EnrichedElastiCache method), 22

get\_related\_cluster\_endpoints() (botoform.enriched.elasticache.EnrichedElastiCache method), 22

get\_related\_cluster\_ids() (botoform.enriched.elasticache.EnrichedElastiCache method), 22

get\_related\_connection\_data() (botoform.enriched.rds.EnrichedRds method), 25

get\_related\_db\_descriptions() (botoform.enriched.rds.EnrichedRds method), 25

get\_related\_db\_endpoints() (botoform.enriched.rds.EnrichedRds method), 25

get\_related\_db\_ids() (botoform.enriched.rds.EnrichedRds method), 25

get\_related\_elb\_descriptions() (botoform.enriched.elb.EnrichedElb method), 23

get\_related\_elb\_names() (botoform.enriched.elb.EnrichedElb method), 23

get\_related\_launch\_config\_descriptions() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_related\_launch\_config\_names() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22

get\_related\_subnet\_group\_descriptions() (botoform.enriched.elasticache.EnrichedElastiCache

method), 22  
 get\_role() (botoform.enriched.vpc.EnrichedVPC method), 27  
 get\_roles() (botoform.enriched.vpc.EnrichedVPC method), 27  
 get\_route\_table() (botoform.enriched.vpc.EnrichedVPC method), 27  
 get\_running\_instances() (botoform.enriched.vpc.EnrichedVPC method), 28  
 get\_security\_group() (botoform.enriched.vpc.EnrichedVPC method), 28  
 get\_subnet() (botoform.enriched.vpc.EnrichedVPC method), 28  
 get\_vgw() (botoform.enriched.vpc.EnrichedVPC method), 28  
 get\_vpc\_by\_name\_tag() (botoform.enriched.vpc.EnrichedVPC method), 28  
 get\_vpn\_gateways() (botoform.enriched.vpc.EnrichedVPC method), 28

## H

hostname (botoform.enriched.instance.EnrichedInstance attribute), 23

## I

id\_human (botoform.enriched.instance.EnrichedInstance attribute), 23  
 id\_to\_human() (in module botoform.util), 19  
 identifiers (botoform.enriched.instance.EnrichedInstance attribute), 23  
 identity (botoform.enriched.instance.EnrichedInstance attribute), 23  
 identity (botoform.enriched.vpc.EnrichedVPC attribute), 28  
 include\_instances() (botoform.enriched.vpc.EnrichedVPC method), 28  
 increase\_indent() (botoform.util.BotoformDumper method), 18  
 instance\_profiles() (botoform.builders.EnvironmentBuilder method), 16  
 instance\_role() (botoform.builders.EnvironmentBuilder method), 16  
 instance\_roles() (botoform.builders.EnvironmentBuilder method), 16  
 instances (botoform.enriched.vpc.EnrichedVPC attribute), 28  
 instances() (in module botoform.plugins.dump), 32

is\_autoscaled (botoform.enriched.instance.EnrichedInstance attribute), 23

## K

key\_names (botoform.enriched.key\_pair.EnrichedKeyPair attribute), 24  
 key\_pairs (botoform.enriched.key\_pair.EnrichedKeyPair attribute), 24  
 key\_pairs() (botoform.builders.EnvironmentBuilder method), 16  
 key\_value\_to\_dict() (in module botoform.util), 19

## L

levels (botoform.util.Log attribute), 18  
 list\_all\_private\_resource\_record\_sets() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 list\_all\_resource\_record\_sets() (botoform.enriched.route53.EnrichedRoute53 method), 25  
 load() (botoform.config.ConfigLoader method), 17  
 load\_balancers() (botoform.builders.EnvironmentBuilder method), 16  
 lock() (botoform.enriched.instance.EnrichedInstance method), 23  
 lock() (in module botoform.plugins.lock), 31  
 lock\_instances() (botoform.enriched.vpc.EnrichedVPC method), 28  
 Log (class in botoform.util), 18

## M

main() (botoform.plugins.create.Create static method), 31  
 main() (botoform.plugins.dump.Dump static method), 32  
 make\_filter() (in module botoform.util), 20  
 make\_tag\_dict() (in module botoform.util), 20  
 map\_filter\_false() (in module botoform.util), 20  
 merge\_pages() (in module botoform.util), 20

## N

name (botoform.enriched.instance.EnrichedInstance attribute), 24  
 name (botoform.enriched.vpc.EnrichedVPC attribute), 28  
 normalize\_sg\_port() (in module botoform.util), 20  
 normalize\_sg\_rules() (in module botoform.util), 20

## O

output\_formatter() (in module botoform.util), 20

## P

private\_zone\_id (botoform.enriched.route53.EnrichedRoute53 attribute), 25  
 private\_zone\_name (botoform.enriched.route53.EnrichedRoute53 attribute), 25

- profile\_name (botoform.util.BotoConnections attribute), 18
- ## R
- reflect\_attrs() (botoform.enriched.instance.EnrichedInstance method), 24
- reflect\_attrs() (botoform.enriched.vpc.EnrichedVPC method), 28
- reflect\_attrs() (in module botoform.util), 20
- refresh\_boto\_connections() (botoform.util.BotoConnections method), 18
- refresh\_private\_zone() (botoform.enriched.route53.EnrichedRoute53 method), 25
- region\_name (botoform.enriched.vpc.EnrichedVPC attribute), 28
- region\_name (botoform.util.BotoConnections attribute), 18
- register\_role\_with\_load\_balancer() (botoform.enriched.elb.EnrichedElb method), 23
- related\_ids() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 29
- reload() (botoform.enriched.instance.EnrichedInstance method), 24
- reload() (botoform.enriched.vpc.EnrichedVPC method), 28
- render() (botoform.config.ConfigLoader method), 17
- render\_string() (botoform.config.ConfigLoader method), 17
- REPL() (in module botoform.plugins.repl), 32
- reset\_master\_passwords() (botoform.enriched.rds.EnrichedRds method), 25
- revoke\_inbound\_rules\_from\_sg() (botoform.enriched.vpc.EnrichedVPC method), 28
- revoke\_outbound\_rules\_from\_sg() (botoform.enriched.vpc.EnrichedVPC method), 28
- revoke\_security\_group\_rules() (botoform.enriched.vpc.EnrichedVPC method), 28
- role (botoform.enriched.instance.EnrichedInstance attribute), 24
- roles (botoform.enriched.vpc.EnrichedVPC attribute), 28
- route\_table\_rules() (botoform.builders.EnvironmentBuilder method), 16
- route\_tables() (botoform.builders.EnvironmentBuilder method), 16
- ## S
- scale\_related\_autoscaling\_group() (botoform.enriched.autoscaling.EnrichedAutoscaling method), 22
- security\_group\_inbound\_rules() (botoform.builders.EnvironmentBuilder method), 16
- security\_group\_outbound\_revoke\_default\_rule() (botoform.builders.EnvironmentBuilder method), 16
- security\_group\_outbound\_rules() (botoform.builders.EnvironmentBuilder method), 16
- security\_group\_rule\_to\_permission() (botoform.builders.EnvironmentBuilder method), 16
- security\_group\_rules() (botoform.builders.EnvironmentBuilder method), 16
- security\_group\_rules\_to\_permissions() (botoform.builders.EnvironmentBuilder method), 16
- security\_groups() (botoform.builders.EnvironmentBuilder method), 16
- security\_groups() (in module botoform.plugins.dump), 32
- services() (botoform.enriched.vpc\_endpoint.EnrichedVpcEndpoint method), 29
- setup\_parser() (botoform.plugins.create.Create static method), 31
- setup\_parser() (botoform.plugins.dump.Dump static method), 32
- setup\_session\_and\_refresh\_connections() (botoform.util.BotoConnections method), 18
- shortname (botoform.enriched.instance.EnrichedInstance attribute), 24
- snake\_to\_camel\_case() (in module botoform.util), 21
- source\_dest\_check\_disable() (botoform.enriched.instance.EnrichedInstance method), 24
- source\_dest\_check\_enable() (botoform.enriched.instance.EnrichedInstance method), 24
- start\_instances() (botoform.enriched.vpc.EnrichedVPC method), 28
- stop\_instances() (botoform.enriched.vpc.EnrichedVPC method), 28
- subnets() (botoform.builders.EnvironmentBuilder method), 16
- ## T
- tag\_dict (botoform.enriched.instance.EnrichedInstance attribute), 24
- tag\_dict (botoform.enriched.vpc.EnrichedVPC attribute), 28
- tag\_filter() (in module botoform.util), 21

`tag_instance_name()` (botofrom.builders.EnvironmentBuilder method), 16

`tag_instance_volumes()` (botofrom.builders.EnvironmentBuilder method), 16

`taggable_resources` (botofrom.enriched.vpc.EnrichedVPC attribute), 28

`tags()` (botofrom.builders.EnvironmentBuilder method), 17

`template_dir` (botofrom.config.ConfigLoader attribute), 17

`terminate()` (botofrom.enriched.vpc.EnrichedVPC method), 28

## U

`unlock()` (botofrom.enriched.instance.EnrichedInstance method), 24

`unlock()` (in module botofrom.plugins.unlock), 32

`unlock_instances()` (botofrom.enriched.vpc.EnrichedVPC method), 28

`update_tags()` (in module botofrom.util), 21

## W

`wait_for_instance_profile()` (botofrom.builders.EnvironmentBuilder method), 17

`wait_for_instance_roles_to_exist()` (botofrom.builders.EnvironmentBuilder method), 17

`wait_for_related_clusters()` (botofrom.enriched.elasticache.EnrichedElastiCache method), 22

`wait_for_related_dbs()` (botofrom.enriched.rds.EnrichedRds method), 25

`wait_until_instances()` (botofrom.enriched.vpc.EnrichedVPC method), 29

`wait_until_status_ok()` (botofrom.enriched.instance.EnrichedInstance method), 24

`write_private_key()` (in module botofrom.util), 21

## Y

`yaml_representers` (botofrom.util.BotoformDumper attribute), 18