
foremast Documentation

Release 3.17.2a9

Gogo DevOps

May 24, 2017

Contents

1	Contents	1
1.1	Getting Started	1
1.2	Pipeline Flow and Examples	3
1.3	Configuration Files	5
1.4	Advance Usages	31
1.5	Infrastructure	33
1.6	AWS Lambda Pipeline	35
1.7	How To Contribute	41
1.8	How To Create Releases	43
1.9	Foremast Modules	44
2	Spinnaker Foremast	93
2.1	Why Foremast?	93
2.2	Foremast Features	93
2.3	Getting Started	93
2.4	More Details	94
3	Indices and tables	97
	Python Module Index	99

Getting Started

- *Quick Start Guide*
 - *Installation*
 - * *Method 1 - Using pip (Preferred)*
 - * *Method 2 - Using git*
 - *Configuration Files*
 - *Running*

This getting started guide will walk through the process of using Foremast to create an application in Spinnaker and dynamically generate a basic Spinnaker pipeline.

Getting started with Foremast consists of the following steps:

1. Setting up configuration files
2. Installing Foremast
3. Setting up the variables
4. Running Foremast

Quick Start Guide

In this section, we will install, configure and run Foremast to create a basic pipeline.

Installation

Setting up the environment

```
$ pip3 install virtualenv
$ virtualenv -p $(which python3) venv
$ source venv/bin/activate
```

Method 1 - Using pip (Preferred)

```
$ pip install foremast
```

Method 2 - Using git

```
$ git clone https://github.com/gogoit/foremast.git
$ cd foremast
$ pip3 install -U .
```

Configuration Files

Create a runway and `.foremast` directory and go into runway directory.

```
$ mkdir runway .foremast
```

Create `pipeline.json` in runway directory

```
{
  "deployment": "spinnaker",
  "env": [ "dev" ]
}
```

Create `application-master-dev.json` in runway directory.

```
{
  "app": {
    "instance_type": "t2.micro"
  },
  "asg": {
    "max_inst": 1,
    "min_inst": 1
  },
  "regions": [
    "us-east-1"
  ]
}
```

Go to `.foremast` directory and create the `foremast.cfg` file.

```
[base]
domain = example.com
envs = dev,prod
regions = us-east-1
gate_api_url = http://gate.example.com:8084
```

You should now see something similar structure.

```
$ tree -a
.
- .foremast
  | - foremast.cfg
- runway
  - application-master-dev.json
  - pipeline.json

2 directories, 3 files
```

Running

Now from within the root directory, run `foremast-pipeline`.

```
$ GIT_REPO=hello PROJECT=world RUNWAY_DIR=runway/ foremast-pipeline
```

This will create an application in Spinnaker named `helloworld` along with a simple pipeline.

Pipeline Flow and Examples

- *Default Pipeline Flow*
- *Custom Pipelines*
- *Example Workflow*

Foremast generates a single pipeline per region. The pipeline is designed to allow deploying to multiple environment with checkpoints between each transition.

Default Pipeline Flow

The below flow can repeat for as many environments as defined in the configs. At Gogo, most applications go through these stages 3 times as we deploy to dev, stage, and production.

1. Configuration
 - This stages defines the Jenkins trigger, property files, and pipeline-wide notifications
2. Bake
 - Bakes an AMI the specified AMI ID
3. Infrastructure Setup [\$env]
 - Calls a Jenkins job to run the `prepare-infrastructure` Foremast command against a specific account.
 - Setups AWS infrastructure such as ELB, IAM roles, S3 bucket, and DNS needed for an application
4. Deploy \$env
 - Uses Spinnaker to create a cluster and server group in specific account.

- The behavior of this stage is largely based on the *application-master-\$account.json* configs.

5. Attach Scaling Policy [\$env]

- If a scaling policy is defined in *application-master-\$account.json*, attaches it to the deployed server group
- If no policy is defined, this stage is excluded

6. Checkpoint \$next-env

- A manual checkpoint stage. This requires human intervention to approve deployment to the next environment.

Stages 3-6 repeat for each environment/account defined in *pipeline.json*.



The default generated pipeline should look like the above image. This is the basic *bake -> infrastructure -> deploy -> checkpoint* pipeline described above.

Custom Pipelines

You can specify an external templates directory in *foremast.cfg / config.py*. Templates in an external directory will need to have the same directory structure and naming as the default templates. if *templates_path* is set in *foremast.cfg / config.py*, Foremast will first see if the file exists there. If not, it will fall back to the provided templates.

If you need to add more stages or change the defaults, this is all possible via external templates. Please the [foremast-templates repo](#) for examples on the templates.

Example Workflow

At Gogo we have a detailed workflow for using Foremast internally. Feel free to copy our workflow or use it as inspiration for your own. You can view all of our internal templates on the [foremast-templates repo](#).



1. the *application-master-\$account.json* and *pipeline.json* are bundled directly with the application code
2. Developer makes a change to one of those configs and pushes to the application's git repository
3. A server-side git hook detects a change and triggers a Jenkins job to run `Foremast prepare-app-pipeline`. This regenerates the application and pipeline in Spinnaker
4. The application artifacts are build using a Jenkins job and stored as an RPM
5. Spinnaker triggers detect a completed Jenkins jobs and starts a new deployment pipeline
 - (a) Bakes an AMI using build RPM
 - (b) Runs a Jenkins job to run `Foremast prepare-infrastructure`. This builds out the AWS ELB, SG, S3 bucket, and IAM roles
 - (c) Runs a Jenkins jobs to tag the effected git repository with AMI info

- (d) Deploys the generated AMI to desired environments
- (e) Runs QE/QA checks against deployed application
- (f) Tags the repository with deployment information
- (g) Attaches defined scaling policies
- (h) Wants for manual judgment before continuing to the next stage

Configuration Files

foremast.cfg / config.py

- *Purpose*
- *Example Configuration*
- *Configuration Locations*
- *Configuration Details*
 - *[base]*
 - * *domain*
 - * *envs*
 - *types*
 - * *regions*
 - * *ami_json_url*
 - * *gitlab_url*
 - * *gate_api_url*
 - * *templates_path*
 - * *default_ec2_securitygroups*
 - * *default_elb_securitygroups*
 - * *gate_client_cert*
 - * *gate_ca_bundle*
 - *[credentials]*
 - * *gitlab_token*
 - * *slack_token*
 - *[whitelists]*
 - * *asg_whitelist*
 - *[formats]*
 - * *domain*
 - * *app*

```
* dns_elb
* s3_bucket
* jenkins_job_name
- [task_timeouts]
* default
* envs
```

Purpose

This configuration holds information necessary for running foremast such as auth tokens, URLs, whitelists etc

Example Configuration

```
; foremast.cfg
[base]
domain = example.com
envs = dev,stage,prod
regions = us-east-1,us-west-2
ami_json_url = http://s3.bucketname.com/ami_lookup.json
git_url = https://git.example.com
gate_api_url = http://gate-api.example.com:8084
templates_path = ../../foremast-templates

[credentials]
gitlab_token = 123token23423343
slack_token = 123slack3203120312

[whitelists]
asg_whitelist = application1,application2

[formats]
app = {project}{repo}
dns_elb = lb-{project}{repo}.{env}.{domain}
s3_bucket = secret-{env}-{project}

[task_timeouts]
default = 120
envs = { "dev" : { "deleteScalingPolicy": 240} }
```

```
# config.py
CONFIG = {
    'base': {
        'domain': 'example.com',
        'envs': 'dev,stage,prod',
        'regions': 'us-east-1,us-west-2',
        'ami_json_url': 'http://s3.bucketname.com/ami_lookup.json',
        'git_url': 'https://git.example.com',
        'gate_api_url': 'http://gate-api.example.com:8084',
        'templates_path': ' ../../foremast-templates',
    },
}
```

```

'credentials': {
  'gitlab_token': '123token23423343',
  'slack_token': '123slack3203120312',
},
'whitelists': {
  'asg_whitelist': 'application1,application2',
},
'formats': {
  'app': '{project}{repo}',
  'dns_elb': 'lb-{project}{repo}.{env}.{domain}',
  's3_bucket': 'secret-{env}-{project}',
},
'timeouts': {
  'default': 120,
  'envs': { 'dev': { 'deleteScalingPolicy': 240 } },
}
}

```

Configuration Locations

Foremast will look in the following locations, in order, for the `foremast.cfg` or `config.py` config file.

1. `./foremast/foremast.cfg`
2. `~/.foremast/.foremast.cfg`
3. `/etc/foremast/foremast/cfg`
4. `./config.py`

Configuration Details

[base]

Sections for base information such as urls and general configurations

domain

The base domain of your applications. Used for generating DNS

Required: Yes

envs

Comma delimited list of environments/applications that will be managed with Foremast

Example: dev, stage, prod

Required: Yes

types

List of foremast managed Pipeline types to allow.

Type: str

Example: ec2, lambda, manual

Default: ec2, lambda

Required: No

regions

Comma delimited list of AWS regions managed by Foremast

Example: us-east-1, us-west-2

Required: Yes

ami_json_url

FQDN of where to query for AMI ID look ups. See [ami-lookup.json](#) for more details

Required: No

gitlab_url

FQDN of gitlab. Will be used for handling API calls to Gitlab

Required: No

gate_api_url

FQDN Of your spinnaker Gate instance. This is where all API calls to Spinnaker will go

Required: Yes

templates_path

Path to custome templates directory. If provided, Foremast will first look in this directory for any templates. This can be an absolute path, or a path relative to where you where you are running the Foremast commands. See [Pipeline Flow and Examples](#) for more details on custom templates.

Required: No

default_ec2_securitygroups

Comma seperated list of EC2 security groups to include for all deployments

Required: No

Example: office, test_sg, example

default_elb_securitygroups

Comma seperated list of ELB security groups to include for all deployments

Required: No

Example: test_sg, example_elb_sg

gate_client_cert

If accessing Gate via x509 certificate authentication, this value provides the local path to the certificate. Only PEM certs are supported at this time (containing both the key and certificate in the PEM).

Required: No

Example: /var/certs/gate-cert.pem

gate_ca_bundle

If accessing Gate via x509 leveraging a custom certificate authority (such as acting as your own CA), this value provides the local path to the CA bundle. It is recommended to use an existing CA Bundle and append your CA certificate to it (<https://certifi.io/en/latest/>)

Required: No

Example: /var/certs/CA.pem

[credentials]

Section for handling credential configurations such as tokens, usernames, and passwords

gitlab_token

Gitlab token used for authentication in Foremast

Required: No

slack_token

Slack token used for authentication when sending Slack messages from Foremast

Required: No

[whitelists]

Sections for configuring whitelist info

asg_whitelist

Comma delimited list of applications to whitelist from ASG rules

Required: No

[formats]

Section handling the naming convention of applications, elb, iam, s3 buckets and other services.

The most common sections are shown. The complete list of sections and defaults are defined by the underlying library `gogo-utils`.

Any of the possible variables below can be used as the value.

- `domain` organization domain
- `env` dev, qa, production, etc
- `project` lowercase git group/organization
- `repo` lowercase git project/repository
- `raw_project` git group/organization
- `raw_repo` git project/repository

`domain`

A string of your organization's domain

Default: example.com

Required: No

`app`

A string of the format of your application

Default: {repo}{project}

Required: No

`dns_elb`

An FQDN of your application's Elastic Load Balancer (ELB)

Default: {repo}.{project}.{env}.{domain}

Required: No

`s3_bucket`

An string of your base S3 bucket name

Default: archaius-{env}

Required: No

`jenkins_job_name`

An string of the format of the application's jenkins job name

Default: {project}_{repo}

Required: No

[task_timeouts]

Section handling customization of task timeouts when communicating with Spinnaker.

Timeouts can vary per environment and per task.

default

The default task timeout value

Default: 120

Required: No

envs

A json object keyed by environment name. Each value should be a json object keyed by task name.

Default: 120

Required: No

pipeline.json

- *Purpose*
- *Example Configuration*
- *Configuration Details*
 - *type*
 - *owner_email*
 - *documentation*
 - *notifications Block*
 - * *email*
 - * *slack*
 - *promote_restrict*
 - *base*
 - *envs*
 - *pipeline_files*
 - *image Block*
 - * *root_volume_size*
 - *lambda Block*
 - * *app_description*

```
* runtime
* handler
* vpc_enabled
- services Block
  * cloudformation
  * cloudwatchlogs
  * dynamodb
  * elasticsearch
  * firehose
  * kinesis
  * lambda
  * s3
  * sdb
  * ses
  * sns
  * sqs
```

Purpose

This configuration file is used for defining pipeline settings that affect the pipeline as a whole, not a specific account/environment.

Example Configuration

```
{
  "type": "ec2",
  "owner_email": "",
  "documentation": "",
  "notifications": {
    "email": "",
    "slack": ""
  },
  "promote_restrict": "none",
  "base": "tomcat8",
  "env": ["stage", "prod"],
  "primary_region": "us-east-1",
  "image": {
    "root_volume_size": 6,
    "builder": "ebs"
  },
  "lambda": {
    "app_description": "default description",
    "runtime": "java8",
    "handler": "main",
    "vpc_enabled": false
  }
}
```



```
    },  
    "pipeline_files": []  
  }  
}
```

Configuration Details

`type`

Specifies what type of pipeline to use for the application.

Default: "ec2"

Options:

- "lambda" - Sets up the AWS Lambda pipeline and infrastructure
- "ec2" - Sets up the AWS EC2 pipeline and infrastructure
- "manual" - Create Pipelines from raw JSON, use with *pipeline_files*.

`owner_email`

The application owners email address. This is not used directly in the pipeline but can be consumed by other tools

Default: null

`documentation`

Link to the applications documentation. This is not used directly in the pipeline but can be consumed by other tools

Default: null

notifications Block

Where to send pipeline failure notifications

`email`

Email address to send pipeline failures (email must be configured in Spinnaker Echo)

Default: null

`slack`

Slack channel to send pipeline failures (Slack must be configured in Spinnaker Echo)

Default: null

`promote_restrict`

Restriction setting for promotions to prod* accounts.

Default: "none"

Options:

- "masters-only" - only masters/owners on a repository can approve deployments
- "members-only" - Any member of a repository can approve deployments
- "none" - No restrictions

`base`

The base AMI to use for baking the application. This can be an alias defined in *ami-lookup.json* or an AMI Id.

Default: "tomcat8"

`envs`

List of accounts that the application will be deployed to. Order matters as it defines the order of the pipeline. The accounts should be named the same as you have them in Spinnaker Clouddriver

Type: List of strings

Default: ["stage", "prod"]

`pipeline_files`

List of JSON files to use for manual *type*.

Type: list

Default: []

`image Block`

Holds settings for the baked image

`root_volume_size`

Defines the root volume size of the resulting AMI in GB

Type: int

Units: Gigabyte

Default: 6

`lambda Block`

Holds settings related to lambda deployments

app_description

Lambda function description

Default: "default description"

runtime

The runtime environment for the Lambda function

Default: "java8"

Options:

- "java8"
- "nodejs"
- "nodejs4.3"
- "python2.7"
- "python3.6"

handler

The function that Lambda calls to begin execution

Default: "main"

vpc_enabled

Whether or not the Lambda function should use a VPC

Type: Boolean

Default: false

services Block

Access to different Cloud Services will be added to an inline Policy for an IAM Role. Keys must match with a corresponding template in `src/foremast/templates/infrastructure/iam/key.json.j2`.

cloudformation

Add CloudFormation access.

Type: bool

Default: false

cloudwatchlogs

Add CloudWatch Logs access. Lambda Functions will automatically have this added.

Type: bool
Default: false

dynamodb

Add DynamoDB access to tables listed.

Type: list
Default: []

elasticsearch

Add ElasticSearch access to domains listed.

Type: list
Default: []

firehose

Add Firehose access to streams listed.

Type: list
Default: []

kinesis

Add Kinesis Streams access to streams listed.

Type: list
Default: []

lambda

Add Lambda access.

Type: bool
Default: false

s3

Add S3 access. You may need to override default templates, see [templates_path](#). Alternatively, you can provide a list of s3 bucket names to be added to the instance profile.

Type: bool -or- list
Default: false

sdb

Add SimpleDB access to SimpleDB Domains listed.

Type: list
Default: []

ses

Add SES access.

Type: bool
Default: false

sns

Add SNS access.

Type: bool
Default: false

sqs

Add SQS access.

Type: bool
Default: false

application-master-\$account.json

- *Purpose*
- *Example Configuration*
- *Configuration Details*
 - *app Block*
 - * *app_description*
 - * *eureka_enabled*
 - * *instance_profile*
 - * *instance_type*
 - * *lambda_environment*
 - *lambda_environment* Keys
 - *lambda_environment* Example
 - * *lambda_memory*
 - * *lambda_role*

- * *lambda_timeout*
- *asg Block*
 - * *hc_type*
 - * *app_grace_period*
 - * *max_inst*
 - * *min_inst*
 - * *ssh_keypair*
 - * *subnet_purpose*
 - * *enable_public_ips*
 - * *scaling_policy*
 - *scaling_policy* Keys
 - *scaling_policy* Example
- *elb Block*
 - * *access_log*
 - *access_log* Keys
 - * *connection_draining_timeout*
 - * *certificate*
 - * *health*
 - *health* Keys
 - * *idle_timeout*
 - * *ports*
 - *ports* Keys
 - *ports* Example
 - * *subnet_purpose*
 - * *target*
- *regions Key*
- *deploy_strategy Key*
- *security_group Block*
 - * *description*
 - * *elb_extras*
 - * *instance_extras*
 - * *ingress*
 - * *egress*
 - * *security_group* Example
- *dns Block*

```
* ttl
- lambda_triggers
```

Purpose

This configuration file holds infrastructure information for \$account. Each AWS account in your pipeline would need a separate application-master-\$account.json file. If your account is named dev, you would want an application-master-dev.json file.

Example Configuration

```
{
  "app": {
    "app_description": null,
    "email": null,
    "eureka_enabled": false,
    "instance_profile": "{{ profile }}",
    "instance_type": "t2.micro",
    "lambda_environment": null,
    "lambda_memory": "128",
    "lambda_role": null,
    "lambda_timeout": "30",
    "canary": false
  },
  "asg": {
    "hc_type": "ELB",
    "hc_grace_period": 180,
    "app_grace_period": 0,
    "max_inst": 3,
    "min_inst": 1,
    "ssh_keypair": null,
    "subnet_purpose": "internal",
    "enable_public_ips": null,
    "provider_healthcheck": {
      "amazon": false
    },
    "scaling_policy": {}
  },
  "elb": {
    "certificate": null,
    "policies": [],
    "listener_policies": [],
    "backend_policies": [],
    "idle_timeout": null,
    "access_log": {},
    "connection_draining_timeout": null,
    "health": {
      "interval": 20,
      "threshold": 2,
      "timeout": 10,
      "unhealthy_threshold": 5
    },
    "i_port": 8080,
    "i_proto": "HTTP",
```

```
    "lb_port": 80,
    "lb_proto": "HTTP",
    "subnet_purpose": "internal",
    "target": "TCP:8080"
  },
  "qe": {
  },
  "regions": [
    "us-east-1"
  ],
  "deploy_strategy": "highlander",
  "security_group": {
    "description": "Auto-Gen SG for {{ app }}",
    "egress": "0.0.0.0/0",
    "elb_extras": [],
    "ingress": {
    },
    "instance_extras": []
  },
  "dns": {
    "ttl": 60,
    "failover_dns": true,
    "region_specific": true
  },
  "lambda_triggers": [],
  "s3": {
    "shared_bucket_master": false,
    "path": "/",
    "bucket_acl": "private",
    "bucket_policy": {},
    "website": {
      "enabled": true,
      "index_suffix": "index.html",
      "error_document": " "
    }
  }
}
```

Configuration Details

app Block

Top level key that contains information on the application and EC2 details

app_description

Describes the application.

Default: null

eureka_enabled

Setting this value to `true` will not create an ELB, DNS record, and set the ASG health check to EC2.

Type: Boolean
Default: false

instance_profile

The instance profile to start EC2 instances with.

Default: "\${stack}_\${app}_profile" - Profile with this name will be created by default. Other profiles need to be created before usage

instance_type

The size/type of the EC2 instance. Uses Standard AWS instance names. See <https://aws.amazon.com/ec2/instance-types/> for details

Default: "t2.micro"

lambda_environment

Environment variables which are passed to the lambda function.

lambda_environment Keys

Variables : Dictionary of environment variables.

Type: Dict
Default: null

lambda_environment Example

```
"environment": {  
  "Variables": {  
    "VAR1": "val1",  
    "VAR2": "val2",  
    "VAR3": "val3"  
  }  
}
```

lambda_memory

The amount of memory to give a Lambda function

Default: "128"
Units: Megabytes

`lambda_role`

Override the default generated IAM Role name.

Default: "\${stack}_\${app}_role"

`lambda_timeout`

The timeout setting for Lambda function

Default: "3600"

Units: Seconds

asg Block

Top level key containing information regarding application ASGs

`hc_type`

ASG Health check type (EC2 or ELB)

Default: "ELB"

Options:

- "ELB"
- "EC2"

`app_grace_period`

App specific health check grace period (added onto default ASG healthcheck grace period) to delay sending of health check requests. This is useful in the event your application takes longer to boot than the default `hc_grace_period` defined in templates. For example, `hc_grace_period` may be 180 seconds, but an app may need a variable amount of time to boot (say 30 seconds extra). This will add 180 + 30 to calculate the overall `hc_grace_period` of 210 seconds.

Default: 0

Units: Seconds

`max_inst`

Maximum number of instances ASG will scale to.

Type: int

Default: 3

`min_inst`

Minimum number of instances your auto-scaling group should have at all times. This is also the default number of instances

Type: int

Default: 1

ssh_keypair

SSH key that your EC2 instances will use. Must already be created in AWS. This replaces the non-functional and deprecated `app_ssh_key` configuration key.

Default: "`{{ account }}_{{ region }}_default`" - `{{ account }}` being the AWS account in the configuration name

subnet_purpose

Determines if the instances should be public (external) or non-public (internal).

Default: "internal"

Options

- "internal"
- "external"

enable_public_ips

Determines if instances in an cluster should have public IPs associated. By default, this is set to null which means it uses default behavior configured for your subnets in your cloud provider.

Type: boolean

Default: null

Options

- true
- false

scaling_policy

Defines scaling policy to attach to ASG. If this block does not exist, no scaling policy will be attached

scaling_policy Keys

`metrics` : The metrics to use for auto-scaling.

Default: "CPUUtilization"

Options:

- "CPUUtilization"
- "NetworkIn"
- "NetworkOut"
- "DiskReadBytes"

`threshold` : Metrics value limit for scaling up

Type: int

period_minutes : Time period to look across for determining if threshold was met

Type: int

Units: Minutes

statistic: Statistic to calculate at the period to determine if threshold was met

Default: "Average"

Options:

- "Average"
- "Maximum"
- "Minimum"
- "Sum"

scaling_policy *Example*

```
"scaling_policy": {
  "metric": "CPUUtilization",
  "threshold": 90,
  "period_minutes": 10,
  "statistic": "Average"
}
```

e1b Block

Top level key for ELB configuration

access_log

Access Log configuration block. Ensure S3 bucket has proper bucket policy to enable writing.

access_log *Keys*

bucket_name : Name of S3 bucket to write access log to

Type: string

Default: Null

bucket_prefix : Prefix to write to in the S3 bucket

Type: string

Default: Null

emit_interval : ELB Access Log write delay

Type: int

Range: 5-60

Units: seconds

Default: Null

`connection_draining_timeout`

Connection Draining Timeout to set on the ELB. This allows existing requests to complete before the load balancer shifts traffic away from a deregistered or unhealthy instance.

Type: int
Range: 1-3600
Units: seconds
Default: Null

`certificate`

Name of SSL certification for ELB. SSL certificate must be uploaded to AWS first

Default: Null

`health`

Health check configuration block

`health Keys`

`interval` : ELB health check interval

Type: int
Units: seconds
Default: 20

`threshold` : Number of consecutive health check successes before declaring EC2 instance healthy.

Default: 2

`timeout` : Health check response timeout

Type: int
Units: seconds
Default: 10

`unhealthy_threshold` : number of consecutive health check failures before declaring EC2 instance unhealthy

Default: 5

`idle_timeout`

Idle Timeout to set on the ELB. This the time, in seconds, that the connection is allowed to be idle (no data has been sent over the connection) before it is closed by the load balancer.

Type: int
Range: 1-3600
Units: seconds
Default: 60

ports

Defines ELB listeners. Expects a list of listeners.

ports Keys

`instance` : The protocol:port of the instance

Default: "HTTP:8080"

`loadbalancer` : the protocol:port of the load balancer

Default: "HTTP:80"

`stickiness` : defines stickiness on ELB; if app, specify `cookie_name`, if elb, specify `cookie_ttl`

Default: None

Supported Types: elb, app

Example:

```
"stickiness": {
  "type": "app",
  "cookie_name": "$cookiename"
}

"stickiness": {
  "type": "elb",
  "cookie_ttl": 300
}
```

`certificate` : The name of the certificate to use if required

Default: null

`listener_policies` : A list of listener policies to associate to an ELB. Must be created in AWS first.

Default: []

Type: List of strings

`backend_policies` : A list of backend server policies to associate to an ELB. Must be created in AWS first.

Default: []

Type: List of strings

Example: ["WebSocket-Proxy-Protocol"]`

ports Example

```

"ports": [
  {
    "instance": "HTTP:8080",
    "loadbalancer": "HTTP:80",
    "stickiness": {
      "type": "app",
      "cookie_name": "cookie"
    }
  },
  {
    "certificate": "my_cert",
    "instance": "HTTP:8443",
    "loadbalancer": "HTTPS:443",
    "listener_policies": ["MyExamplePolicy"],
    "stickiness": {
      "type": "elb",
      "cookie_name": 300
    }
  }
]

```

subnet_purpose

Determines if the load balancer should be public (external) or non-public (internal).

Default: "internal"

Options:

- "internal"
- "external"

target

The check the ELB will use to validate application is online.

Default: "TCP:8080"

regions Key

List of AWS regions that application will be deployed to.

Type: List of strings

Default: ["us-east-1"]

deploy_strategy Key

Spinnaker strategy to use for deployments.

Default: "highlander"

Options:

- "highlander" - destroy old server group
- "redblack" - disables old server group but do not destroy

security_group Block

Hold configuration for creating application specific security group

description

Description of the security group. Used in AWS for creation

Default: "Auto-Gen SG for {{ app }}"

elb_extras

A list of extra security groups to assign to ELB

Type: List of strings

Default: []

Example: ["all_access", "test_sg"]`

instance_extras

A list of extra security groups to assign to each instance

Type: List of strings

Default: []

Example: ["all_access", "test_sg"]`

ingress

Provides a list of other security groups and ports to allow inbound access to application

egress

Provides info about outbound access from application

Default: "0.0.0.0/0"``

security_group Example

You can reference SG by name or by cidr block, you can also specify cross account SG by name by referring to the spinnaker environment name. To see an example of this see below:

```
"security_group": {
  "ingress": {
    "examplesecuritygroupname": [
      { "start_port": 80, "end_port": 80, "protocol": "tcp" },
```



```

        { "start_port": 443, "end_port": 443, "protocol": "tcp" },
        { "start_port": 443, "end_port": 443, "protocol": "tcp", "env": "prod" },
    ],
    "192.168.100.0/24": [
        { "start_port": 80, "end_port": 80, "protocol": "tcp" }
    ]
},
"egress": {
    "192.168.100.0/24": [
        { "start_port": 80, "end_port": 80, "protocol": "tcp" }
    ]
}
}

```

dns Block

Top level key for dns settings

ttl

Defines DNS TTL for generated DNS records

Type: int
Units: seconds
Default: 60

lambda_triggers

A list of all events to trigger a Lambda function. See *Lambda Triggers and Events* for details

Type: List
Default: []

AWS Credentials

- *Purpose*
- *Example Configuration*
- *Configuration Location*
- *Configuration Details*

Purpose

This is how AWS credentials are stored for usage with Foremast. All AWS calls outside of Spinnaker use Boto3 so standard Boto3 locations work but account/environment must be specified.

Example Configuration

```
[build]
aws_access_key_id = XXXXXXXXXXXXXXXXXXXXXXXXXXXX
aws_secret_access_key = yyyyyyyxxxxxxxxxyyyyyyyyyyyxxxxxxx

[dev]
aws_access_key_id = AAAAAAAAAAAAAAAAAAAAAAAAAA
aws_secret_access_key = bbbbbbbbaaaaaaaaaabbbbbbbbaaaaa

[stage]
aws_access_key_id = TTTTTTTTTTTTTTTTTTTTTTTT
aws_secret_access_key = ssssssssstttttttttttsssssssss
```

Configuration Location

Foremast just uses Boto3 which will look at `~/.aws/credentials` for the `credentials` file.

Configuration Details

This is a standard Boto3 `credentials` file. You can read more about it on the [Boto3 docs](#). The important part is that each account/environment that Foremast is managing has a distinct section in `credentials`.

ami-lookup.json

- *Purpose*
- *Example Json*
- *Json Location*

Purpose

This json file is used as an AMI ID look up table for each region. It is used during the bake stage of Spinnaker deployments in order to determine the base AMI ID to use for baking.

Example Json

```
{
  "us-east-1": {
    "origin": "ami-xxxx",
    "origin_default": "ami-xxxx",
    "origin_fedora": "ami-xxxx",
    "origin_amazon": "ami-xxxx",
    "origin_ubuntu": "ami-xxxx",
    "origin_debian": "ami-xxxx",
    "origin_testing": "ami-xxxx",
  }
  "us-west-2": {
    "origin": "ami-xxxx",
```

```

"origin_default": "ami-xxxx",
"origin_fedora": "ami-xxxx",
"origin_amazon": "ami-xxxx",
"origin_ubuntu": "ami-xxxx",
"origin_debian": "ami-xxxx",
"origin_testing": "ami-xxxx",
}
}

```

Json Location

Foremast will look for this information at `ami_json_url` defined in `foremast.cfg / config.py`. For example, you can host the file named `ami-lookup.json` in an S3 bucket and then set `ami_json_url = http://s3bucketurl.com/ami-lookup.json`.

You can host this file anywhere as long as an HTTP GET will return the JSON and a 2XX.

Advance Usages

- *Environment Variables*
- *Pipeline Configs*
- *Running Foremast*
 - *Method 1*
 - *Method 2*
- *Next Steps*

This section will show many advance usages of Foremast.

Environment Variables

These are environment variables used when executing Foremast

`TRIGGER_JOB`: The name of the Jenkins job that Spinnaker should look for as a trigger
`APPNAME`: The full name of your application in Spinnaker. `${GIT_REPO}${PROJECT}` is default
`EMAIL`: Email address associated with application in Spinnaker
`PROJECT`: The namespace or group of the application being set up
`GIT_REPO`: The name of the repo in the above namespace/group
`RUNWAY_DIR`: Path to the `pipeline.json` and `application-master-$account.json` files created above

Pipeline Configs

The `pipeline.json` and `application-master-$account.json` are critical files that determine on how each application in the pipeline will work. We recommend keeping these files in the same repository as your application but as long as they are on the same local machine as the Foremast runner they can be used.

In `~/runway` create a file `pipeline.json` with the contents:

```
{
  "deployment": "spinnaker",
  "env": [ "account1", "account2" ]
}
```

In the same `~/runway` directory, create a file `application-master-$account.json` where `$account` is the same name as an account in your AWS credentials file and in your `env` list in `pipeline.json`. This file can be empty and it will just use the defaults provided at [application-master-\\$account.json](#).

Note: You will need an `application-master-$account.json` config for each `$account` that you are deploying to.

See [pipeline.json](#) and [application-master-\\$account.json](#) for all configuration options.

Running Foremast

After setting up all of the configs there are a couple of ways to run Foremast components. You can use our bundled CLI endpoints that look at environment variables, or you can call each individual component with appropriate arguments on the CLI

Both methods will generate the same outcome. An application created in Spinnaker and a pipeline generated based on the configs.

Method 1

This is our recommended method and how we internally at Gogo run Foremast. You need to first set the environment variables from above.

With the environment variables defined, you can simply run the command `foremast-pipeline` from the command line. This will create the Application in Spinnaker as well as generate a base pipeline.

Method 2

This method is more explicit and requires calling multiple Foremast components to create the configs, create the application, and generate the pipeline:

```
create-configs -o ./raw.properties -g ${PROJECT}/${GIT_REPO} -r ${RUNWAY_DIR}
create-app -a ${APPNAME} --email ${EMAIL} --project ${PROJECT} --repo ${GIT_REPO}
create-pipeline -a ${APPNAME} --triggerjob ${TRIGGER_JOB}
```

Next Steps

Take a look at the [Infrastructure](#) docs for details on the necessary Jenkins jobs.

This is only the tip of what Foremast can do. It also has functionality for creating scaling policies, setting up AWS infrastructure (elbs, security groups, iam policies, s3 buckets), sending slack notifications, and destroying old infrastructure. Take a look at our internal workflow docs for more detail on how Foremast is used at Gogo.

Infrastructure

- *Spinnaker*
 - *Spinnaker Component Versions*
- *AWS*
- *Jenkins*
 - *Necessary Jenkins Jobs*
- *Gitlab*

Spinnaker

- Foremast assumes that Spinnaker is already setup. Please see the [Spinnaker documentation](#) for assistance
- Requires connectivity to the Gate component of Spinnaker. Foremast also supports x509 authentication on Gate.
- Assumes AWS EBS is used for Packer bakes in Spinnaker Rosco

Spinnaker Component Versions

Below are the Spinnaker component versions that we use internally at Gogo and that Foremast has been tested against:

- Gate: 2.70.0
- Clouddriver: 1.286.0
- Deck: 2.711.0
- Igor: 1.50.0
- Echo: 1.108.0
- Front50: 1.38.0
- Spinnaker: 0.50.0
- Rosco: 0.42.0
- Orca: 1.168.0

If you have any issues with Foremast at other Spinnaker versions please file an issue (or pull request).

AWS

Foremast only works with AWS (for now). Below are the AWS requirements:

- Foremast IAM Access:
 - Will need credentials set up in a Boto3 configuration file. See [AWS Credentials](#) for details.
 - IAM user or role will need the following permissions:
 - * S3: View, create and delete buckets.
 - * IAM: View, create and delete roles, users, and policies.

- * Route53: View, create, and delete DNS records.
- Everything else, such as ELBs and security groups, are handled through Spinnaker.
- VPC Subnets
 - If new subnets are being setup, follow the [Spinnaker AWS Setup guide](#).
 - If using existing subnets add an `immutable_metadata` tag.
 - * Example `immutable_metadata` tag: `{"purpose": "external", "target": "elb"}`
 - * The "purpose" key will dictate how this appears in Spinnaker.
 - Needs to be "internal" or "external" in order to properly work with Foremast

Jenkins

Foremast takes advantage of the Spinnaker Jenkins stage. In order for the Foremast generated pipeline to work you will need the following:

- Jenkins configuration named “JenkinsCI” in Spinnaker Igor
 - Example Igor config:

```
jenkins:
  Masters:
  -
    name: 'JenkinsCI' # The display name for this server
    address: 'http://jenkinsci.example.com'
    username: 'spinnaker'
    password: 'password'
```

Necessary Jenkins Jobs

The default generated pipeline requires a couple of Jenkins jobs to be setup in order to run.

- `pipes-pipeline-prepare`
 - Runs Foremast `prepare-infrastructure` during the “Infrastructure Setup” pipeline stage
 - Requires the following string variables
 - * `PROJECT`
 - * `GIT_REPO`
 - * `ENV`
 - * `REGION`
 - Example Shell after cloning Foremast:

```
virtualenv -p python3 venv
. venv/bin/activate
pip install -U --quiet .
prepare-infrastructure
```

- `pipes-scaling-policy`

- Runs Foremast `create-scaling-policy` for attaching a scaling policy if defined.
- Only necessary if you plan on attaching scaling policies
- Requires the following string variables
 - * PROJECT
 - * GIT_REPO
 - * ENV
 - * REGION
- Example Shell after cloning Foremast:

```
virtualenv -p python3 venv
. venv/bin/activate
pip install -U --quiet .
create-scaling-policy
```

Gitlab

Gitlab is not required for Spinnaker but if it is already part of your infrastructure you can have Foremast directly look up the `pipeline.json` and `application-master-$account.json` files. You will need to get the Gitlab Token of a user that has permissions to the desired repository and set them in your `foremast.cfg / config.py`.

AWS Lambda Pipeline

Lambda Triggers and Events

- *Purpose*
- *Example Configuration*
- *Configuration Details*
 - *type*
 - *S3 Event*
 - * *bucket*
 - * *events*
 - * *prefix*
 - * *suffix*
 - *SNS Event*
 - * *topic*
 - *Cloudwatch Event*
 - * *schedule*
 - * *rule_name*

```
* rule_description
- Cloudwatch Log Event
  * log_group
  * filter_name
  * filter_pattern
- API Gateway Event
  * api_name
  * resource
  * method
```

Purpose

Foremast supports multiple Lambda events. These are configured in the *application-master-\$account.json* config and set as a list under the *lambda_triggers* key.

Example Configuration

This example would go in the *application-master-\$account.json* configuration file.

```
{
  "lambda_triggers": [
    {
      "type": "s3",
      "bucket": "app-bucket-dev",
      "events": [
        "s3:ObjectCreated:*"
      ],
      "prefix": "",
      "suffix": ""
    },
    {
      "type": "sns",
      "topic": "app-dns-dev"
    },
    {
      "type": "cloudwatch-event",
      "schedule": "rate(5 minutes)",
      "rule_name": "app cron - 5min",
      "rule_description": "triggers lambda function every five minutes"
    },
    {
      "type": "cloudwatch-logs",
      "log_group": "/aws/lambda/awslimit_test",
      "filter_name": "Trigger lambda on every WARNING message",
      "filter_pattern": ""
    },
    {
      "type": "api-gateway",
      "api_name": "lambdatest-api",
      "resource": "/index",
    }
  ]
}
```



```

    "method": "GET"
  }
]
}

```

Configuration Details

type

Specifies what type of Lambda event/trigger to use. This needs to be set for all events.

Options:

- "s3" - S3 Lambda trigger
- "sns" - SNS Lambda trigger
- "cloudwatch-event" - Cloudwatch event Lambda trigger
- "cloudwatch-logs" - Cloudwatch logs event Lambda trigger
- "api-gateway" - API Gateway Lambda trigger

Required: True

S3 Event

A Lambda trigger on S3 bucket actions.

bucket

The bucket of the event to monitor.

Required: True

events

The S3 event to trigger the lambda function from.

Type: List

Required: True

Example: ["s3:ObjectCreated:*", "s3:ObjectRemoved:Delete"]

prefix

Sets up a prefix filter on S3 bucket events.

Required: False

Example: "logs/"

suffix

Sets up a suffix filter on s3 bucket events.

Required: False

Example: "jpg"

SNS Event

A Lambda trigger on SNS topic events.

topic

The SNS topic name to monitor for events.

Required: True

Cloudwatch Event

A Cloudwatch Scheduled event for Lambda triggers.

schedule

The rate or cron string to trigger the Lambda function.

Required: True

Examples:

- "rate(5 minutes) "
- "cron(0 17 ? * MON-FRI *) "

rule_name

The name of the cloudwatch rule being created.

Required: False

Default: "{app_name}+{schedule} "

rule_description

Description of the rule being created.

Required: False

Cloudwatch Log Event

A lambda event that triggers off a Cloudwatch log action.

log_group

The name of the log group to monitor.

Required: True

Example: `"/aws/lambda/test_function"`

filter_name

The name of the filter on log event.

Required: True

filter_pattern

The pattern to look for in the `log_group` for triggering a Lambda function.

Required: True

Example: `"warning"`

API Gateway Event

Sets up an API Gateway event to trigger a lambda function.

api_name

The name of an existing API Gateway. If not provided, an API will be created.

Required: False

Default: `{app_name}`

resource

The API resource to tie the Lambda function to.

Required: True

Example: `"/test"`

method

The API Method to trigger the Lambda function.

Required: True

Example: `"GET"`

Lambda Pipeline

- [Overview](#)
- [Lambda Specific Setup](#)
- [Lambda Pipeline Example](#)

Overview

Foremast supports the ability to setup Lambda infrastructure and build a pipeline around Lambda deployments. This was designed to be very similar to the default EC2 pipeline. It requires the same configuration files and general setup.

Lambda Specific Setup

1. Look at the [Getting Started](#) guide for basic setup. The Lambda process will be very similar
2. Look at the *lambda Block* configurations in *pipeline.json* and *application-master- $\$$ account.json*.
3. In *pipeline.json* set "type" : "lambda" in order for Foremast to treat the application as a Lambda function.
4. Setup the desired Lambda triggers. See [Lambda Triggers and Events](#) for details.

Lambda Pipeline Example

1. Generate a ZIP artifact of your desired Lambda function
2. Trigger Spinnaker Lambda pipeline
3. Spinnaker runs “Infrastructure Setup Lambda”
 - (a) Sets up default function
 - (b) Sets up event triggers
 - (c) Sets up IAM Roles
 - (d) Sets up security groups
4. Spinnaker runs a “Deploy Lambda” stage
 - (a) This stage uploads the ZIP artifact to the created Lambda function
5. Manual Judgement checkpoint for deploying to the next environment
6. Repeat steps 3-5 for each desired environment



How To Contribute

- *Getting Started*
 - *Commits*
 - *Branches*
 - *Documentation*
 - *Testing*
- *Code Submission*
 - *Code Improvement*
 - *Code Submission*
 - *Code Review*
 - *Code Acceptance*

Contributions to Foremast are welcome.

Getting Started

Commits

Follow [semantic commits](#) to make `git log` a little easier to follow.

chore something just needs to happen, e.g. versioning

docs documentation pages in `_docs/` or `docstrings`

feat new code in `src/`

fix code improvement in `src/`

refactor code movement in `src/`

style aesthetic changes

test test case modifications in `test/`

Examples commit messages:

- chore: v10.0
- docs: Add configuration setting
- feat: Create Lambda function
- fix: Retry upload on failure
- refactor: Extract duplicate code
- style: isort, YAPF
- test: Coverage around add permissions

Branches

Use slash convention with the same leaders as *Commits*, e.g.:

- chore/v10.0
- docs/configs
- feat/lambda
- fix/deadlock
- refactor/debug_util
- style/lambda_whitespace
- test/lambda_permission

Documentation

- Use reStructuredText for docstrings and documentation
- For docstrings, follow [Example Google Style Python Docstrings](#)
- For documentation pages, follow the strong guidelines from Python with [Documenting Python](#)

Note:

- Use `.rst` for regular pages
 - Use `.rest` for pages included using `.. include:: file.rest` (fixes a Sphinx issue that thinks references are duplicated)
-

Testing

Run any unit tests available in `./tests/`.

```
virtualenv venv
source ./venv/bin/activate
pip install -U -r requirements-dev.txt

tox
# OR
./runtests.py
```

Code Submission

Code Improvement

1. See if an [Issue](#) exists
 - Comment with any added information to help the discussion
2. Create an [Issue](#) if needed

Code Submission

1. See if a [Pull Request](#) exists
 - Add some comments or review the code to help it along
 - Don't be afraid to comment when logic needs clarification
2. Create a Fork and open a [Pull Request](#) if needed

Code Review

- Anyone can review code
- Any [Pull Request](#) should be closed or merged within a week

Code Acceptance

Try to keep history as linear as possible using a *rebase* merge strategy.

1. One thumb up at minimum, two preferred
2. Request submitter to *rebase* and resolve all conflicts

```
# Update `master`
git checkout master
git pull

# Update `feat/new` Branch
git checkout feat/new
git rebase master

# Update remote Branch and Pull Request
git push -f
```

3. Merge the new feature

```
# Merge `feat/new` into `master`
git checkout master
git merge --ff-only feat/new
git push
```

4. Delete merged Branch

How To Create Releases

Setup

Add the following to `~/.pypirc` file

```
[distutils]
index-servers =
    pypi

[pypi]
repository = https://pypi.python.org/pypi
```

```
username = username
password = xxxyyyyzzz
```

Upload Release

When releasing a new version, the following needs to occur:

1. Add version Tag

```
git tag -a v#.#.#
git push --tags
```

2. Ensure all test via `tox` pass
3. Generate and upload the package

```
python setup.py bdist_wheel upload -r pypi
```

Foremast Modules

foremast package

Subpackages

foremast.app package

Submodules

foremast.app.__main__ module

CLI entry point for creating a Spinnaker application.

Help: `python -m src.foremast.app -h`

`foremast.app.__main__.main()`

Entry point for creating a Spinnaker application.

foremast.app.create_app module

Module for creating an application in spinnaker.

Looks to see if the application exists, and if not creates the application.

```
class foremast.app.create_app.SpinnakerApp(app=None, email=None, project=None,
                                           repo=None)
```

Bases: `object`

Class to manage and create Spinnaker applications

Parameters

- `app` (*str*) – Application name.

- **email** (*str*) – Email associated with application.
- **project** (*str*) – Git namespace or project group
- **repo** (*str*) – Repository name

appinfo

dict – A dictionary containing the provided arguments

appname

str – The name of the application.

create_app()

Send a POST to spinnaker to create a new application with class variables.

Raises `AssertionError` – Application creation failed.

get_accounts(provider='aws')

Get Accounts added to Spinnaker.

Parameters **provider** (*str*) – What provider to find accounts for.

Returns list of dicts of Spinnaker credentials matching `_provider_`.

Return type `list`

Raises `AssertionError` – Failure getting accounts from Spinnaker.

Module contents

Package for creating a Spinnaker application.

foremast.autoscaling_policy package

Submodules

foremast.autoscaling_policy.__main__ module

Deletes existing scaling policy on application and then creates new Autoscaling Policy based on provided configurations

Help: ``python -m src.foremast.autoscaling_policy -h``

`foremast.autoscaling_policy.__main__.main()`

CLI entrypoint for scaling policy creation

foremast.autoscaling_policy.create_policy module

Manages AWS scaling policies in Spinnaker. Can find, create, and delete.

This module also creates an inverse policy for scaling down

```
class foremast.autoscaling_policy.create_policy.AutoScalingPolicy (app='',
                                                                    prop_path='',
                                                                    env='',      re-
                                                                    gion='')
```

Bases: `object`

Manages scaling policies in Spinnaker

Parameters

- **app** (*str*) – Application name
- **prop_path** (*str*) – Path of rendered property files
- **env** (*str*) – Environment/Account to add policy to
- **region** (*str*) – AWS region for policy

log

str – Logger name

settings

dict – Properties imported from prop_path

create_policy ()

Wrapper function. Gets the server group, sets sane defaults, deletes existing policies, and then runs self.prepare_policy_template for scaling up and scaling down policies. This function acts as the main driver for the scaling policy creation process

delete_existing_policy (scaling_policy, server_group)

Given a scaling_policy and server_group, deletes the existing scaling_policy. Scaling policies need to be deleted instead of upserted for consistency.

Parameters

- **scaling_policy** (*json*) – the scaling_policy json from Spinnaker that should be deleted
- **server_group** (*str*) – the affected server_group

get_all_existing (server_group)

Finds all existing scaling policies for an application

Returns List of all existing scaling policies for the application

Return type scalingpolicies (*list*)

get_server_group ()

Finds the most recently deployed server group for the application. This is the server group that the scaling policy will be applied to.

Returns Name of the newest server group

Return type server_group (*str*)

prepare_policy_template (scaling_type, period_sec, server_group)

Renders scaling policy templates based on configs and variables. After rendering, POSTs the json to Spinnaker for creation.

Parameters

- **scaling_type** (*str*) – scale_up or scaling_down. Type of policy
- **period_sec** (*int*) – Period of time to look at metrics for determining scale
- **server_group** (*str*) – The name of the server group to render template for

Module contents

Package for managing the creation and attachment of AWS scaling policies

foremast.awslambda package

Subpackages

foremast.awslambda.api_gateway_event package

Submodules

foremast.awslambda.api_gateway_event.__main__ module

Create API Gateway events and integration.

```
foremast.awslambda.api_gateway_event.__main__.main()
    Create any API Gateway event related resources.
```

foremast.awslambda.api_gateway_event.api_gateway_event module

```
class foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway (app='',
                                                                           env='',
                                                                           re-
                                                                           gion='',
                                                                           rules={},
                                                                           prop_path='')
```

Bases: `object`

Class to handle API Gateway and Lambda integration.

Parameters

- **app** (*str*) – Application Name
- **env** (*str*) – Environment/account for deployments
- **region** (*str*) – AWS Region
- **rules** (*dict*) – Trigger settings
- **prop_path** (*str*) – Path to the raw.properties.json

add_integration_response ()

Add an intergation response to the API for the lambda integration.

add_lambda_integration ()

Attach lambda found to API.

add_permission ()

Add permission to Lambda for the API Trigger.

attach_method (*resource_id*)

Attach the defined method.

create_api ()

Create the REST API.

create_api_deployment ()

Create API deployment of ENV name.

create_api_key ()

Create API Key for API access.

create_resource (*parent_id*='')

Create the specified resource.

Parameters **parent_id** (*str*) – The resource ID of the parent resource in API Gateway

find_api_id ()

Given API name, find API ID.

find_resource_id ()

Given a resource path and API Id, find resource Id.

generate_uris ()

Generate several lambda uris.

setup_lambda_api ()

A wrapper for all the steps needed to setup the integration.

update_api_mappings ()

Create a cname for the API deployment.

Module contents

foremast.awslambda.cloudwatch_event package

Subpackages

foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event package

Submodules

foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.__main__ module

CLI entry point for DNS cleanup.

`foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.__main__.main()`

Destroy Cloudwatch event

foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.destroy_cloudwatch_event module

Destroy any DNS records.

`foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.destroy_cloudwatch_event.dest`

Destroy Cloudwatch event subscription.

Parameters

- **app** (*str*) – Spinnaker Application name.
- **env** (*str*) – Deployment environment.
- **region** (*str*) – AWS region.

Returns True upon successful completion.

Return type `bool`

Module contents

Submodules

`foremast.awslambda.cloudwatch_event.__main__` module

Create Cloudwatch events.

```
foremast.awslambda.cloudwatch_event.__main__.main()
    Create any Cloudwatch events related Resources.
```

`foremast.awslambda.cloudwatch_event.cloudwatch_event` module

```
foremast.awslambda.cloudwatch_event.cloudwatch_event.create_cloudwatch_event(app_name,
                                                                              env,
                                                                              re-
                                                                              gion,
                                                                              rules)
```

Create cloudwatch event for lambda from rules.

Parameters

- **app_name** (*str*) – name of the lambda function
- **env** (*str*) – Environment/Account for lambda function
- **region** (*str*) – AWS region of the lambda function
- **rules** (*dict*) – Trigger rules from the settings

Module contents

`foremast.awslambda.cloudwatch_log_event` package

Subpackages

`foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event` package

Submodules

`foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.__main__` module

CLI entry point for Cloudwatch log subscription cleanup.

```
foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.__main__.main()
    Destroy Cloudwatch log event.
```

foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.destroy_cloudwatch_log_event module

Destroy any cloudwatch log events.

```
foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.destroy_cloudwatch_log_event
```

Destroy Cloudwatch log event.

Parameters

- **app** (*str*) – Spinnaker Application name.
- **env** (*str*) – Deployment environment.
- **region** (*str*) – AWS region.

Returns True upon successful completion.

Return type bool

Module contents

Submodules

foremast.awslambda.cloudwatch_log_event.__main__ module

Create Cloudwatch log event subscription.

```
foremast.awslambda.cloudwatch_log_event.__main__.main()
    Create Cloudwatch log event subscription for Lambda.
```

foremast.awslambda.cloudwatch_log_event.cloudwatch_log_event module

```
foremast.awslambda.cloudwatch_log_event.cloudwatch_log_event.create_cloudwatch_log_event (ap
en
re
gi
ru
```

Create cloudwatch log event for lambda from rules.

Parameters

- **app_name** (*str*) – name of the lambda function
- **env** (*str*) – Environment/Account for lambda function
- **region** (*str*) – AWS region of the lambda function
- **rules** (*str*) – Trigger rules from the settings

Module contents

foremast.awslambda.s3_event package

Subpackages

foremast.awslambda.s3_event.destroy_s3_event package

Submodules

foremast.awslambda.s3_event.destroy_s3_event.__main__ module

CLI entry point for destroying lambda S3 event.

```
foremast.awslambda.s3_event.destroy_s3_event.__main__.main()  
    Destroy S3 event of a bucket
```

foremast.awslambda.s3_event.destroy_s3_event.destroy_s3_event module

Destroy S3 events.

```
foremast.awslambda.s3_event.destroy_s3_event.destroy_s3_event.destroy_s3_event(app,  
                                                                              env,  
                                                                              re-  
                                                                              gion)
```

Destroy S3 event.

Parameters

- **app** (*str*) – Spinnaker Application name.
- **env** (*str*) – Deployment environment.
- **region** (*str*) – AWS region.

Returns True upon successful completion.

Return type bool

Module contents

Submodules

foremast.awslambda.s3_event.__main__ module

Create Lambda S3 event.

```
foremast.awslambda.s3_event.__main__.main()  
    Create Lambda S3 event.
```

foremast.awslambda.s3_event.s3_event module

```
foremast.awslambda.s3_event.s3_event.create_s3_event(app_name, env, region, rules)  
    Create S3 lambda event from rules.
```

Parameters

- **app_name** (*str*) – name of the lambda function

- **env** (*str*) – Environment/Account for lambda function
- **region** (*str*) – AWS region of the lambda function
- **rules** (*str*) – Trigger rules from the settings

Module contents

foremast.awslambda.sns_event package

Subpackages

foremast.awslambda.sns_event.destroy_sns_event package

Submodules

foremast.awslambda.sns_event.destroy_sns_event.__main__ module

CLI entry point destroying Lambda SNS event subscription.

```
foremast.awslambda.sns_event.destroy_sns_event.__main__.main()
    Destroy Lambda SNS event subscription.
```

foremast.awslambda.sns_event.destroy_sns_event.destroy_sns_event module

```
foremast.awslambda.sns_event.destroy_sns_event.destroy_sns_event.destroy_sns_event (app_name,
                                                                                       env,
                                                                                       re-
                                                                                       gion)
```

Destroy all Lambda SNS subscriptions.

Parameters

- **app_name** (*str*) – name of the lambda function
- **env** (*str*) – Environment/Account for lambda function
- **region** (*str*) – AWS region of the lambda function

Returns True if subscription destroyed successfully

Return type boolean

Module contents

Submodules

foremast.awslambda.sns_event.__main__ module

Create Lambda SNS event subscription

```
foremast.awslambda.sns_event.__main__.main()
    Create Lambda SNS event subscription.
```


foremast.awslambda.sns_event.sns_event module

`foremast.awslambda.sns_event.sns_event.create_sns_event` (*app_name*, *env*, *region*, *rules*)

Create SNS lambda event from rules.

Parameters

- **app_name** (*str*) – name of the lambda function
- **env** (*str*) – Environment/Account for lambda function
- **region** (*str*) – AWS region of the lambda function
- **rules** (*str*) – Trigger rules from the settings

Module contents

Submodules

foremast.awslambda.__main__ module

CLI entry point for aws events creation.

Help: `python -m src.foremast.awslambdaevent -h`

`foremast.awslambda.__main__.main()`

Create Lambda events.

foremast.awslambda.awslambda module

class `foremast.awslambda.awslambda.LambdaFunction` (*app*, *env*, *region*, *prop_path*)

Bases: `object`

Manipulate Lambda function.

create_alias ()

Create lambda alias with env name and points it to \$LATEST.

create_function (*vpc_config*)

Create lambda function, configures lambda parameters.

We need to upload non-zero zip when creating function. Uploading hello_world python lambda function since AWS doesn't care which executable is in ZIP.

Parameters **vpc_config** (*dict*) – Dictionary of SubnetIds and SecurityGroupsIds for using a VPC in lambda

create_lambda_function ()

Create or update Lambda function.

update_alias ()

Update lambda alias to point to \$LATEST.

update_function_configuration (*vpc_config*)

Update existing Lambda function configuration.

Parameters **vpc_config** (*dict*) – Dictionary of SubnetIds and SecurityGroupsIds for using a VPC in lambda

foremast.awslambda.awslambdaevent module

Create Lambda event triggers.

```
class foremast.awslambda.awslambdaevent.LambdaEvent (app=None, env=None, re-  
gion=None, prop_path=None)
```

Bases: `object`

Manipulate Lambda events.

```
create_lambda_events ()
```

Create all defined lambda events for an lambda application.

Module contents

foremast.configs package

Submodules

foremast.configs.__main__ module

CLI entrypoint to Application Configuration preparer.

Help: `python -m src.foremast.configs -h`

```
foremast.configs.__main__.main ()
```

Append Application Configurations to a given file in multiple formats.

foremast.configs.outputs module

Write output files for configurations.

```
foremast.configs.outputs.convert_ini (config_dict)
```

Convert `_config_dict_` into a list of INI formatted strings.

Parameters `config_dict` (*dict*) – Configuration dictionary to be flattened.

Returns (list) Lines to be written to a file in the format of `KEY1_KEY2=value`.

```
foremast.configs.outputs.write_variables (app_configs=None, out_file='', git_short='')
```

Append `_application.json_configs` to `_out_file_`, `.exports`, and `.json`.

Variables are written in INI style, e.g. `UPPER_CASE=value`. The `.exports` file contains `'export'` prepended to each line for easy sourcing. The `.json` file is a minified representation of the combined configurations.

Parameters

- **app_configs** (*dict*) – Environment configurations from `_application.json_files`, e.g. `{'dev': {'elb': {'subnet_purpose': 'internal'}}}`.
- **out_file** (*str*) – Name of INI file to append to.
- **git_short** (*str*) – Short name of Git repository, e.g. `forrest/core`.

Returns Configuration equivalent to the JSON output.

Return type `dict`

foremast.configs.prepare_configs module

Prepare the Application Configurations.

```
foremast.configs.prepare_configs.process_git_configs(git_short='')  
    Retrieve _application.json_files from GitLab.
```

Parameters `git_short` (*str*) – Short Git representation of repository, e.g. forrest/core.

Returns Configurations stored for each environment found.

Return type `collections.defaultdict`

```
foremast.configs.prepare_configs.process_runway_configs(runway_dir='')  
    Read the _application.json_files.
```

Parameters `runway_dir` (*str*) – Name of runway directory with app.json files.

Returns Configurations stored for each environment found.

Return type `collections.defaultdict`

Module contents

Package for assembling `foremast` configuration files.

Retrieve the necessary configuration files from GitLab, merge with the default templates, and output a master `settings` file to be consumed by other `foremast` modules.

foremast.dns package

Subpackages

foremast.dns.destroy_dns package

Submodules

foremast.dns.destroy_dns.__main__ module

CLI entry point for DNS cleanup.

```
foremast.dns.destroy_dns.__main__.main()  
    Destroy any DNS related resources of an application
```

Records in any Hosted Zone for an Environment will be deleted.

foremast.dns.destroy_dns.destroy_dns module

Destroy any DNS records.

```
foremast.dns.destroy_dns.destroy_dns.destroy_dns(app='', env='dev', **_)  
    Destroy DNS records.
```

Parameters

- `app` (*str*) – Spinnaker Application name.

- **env** (*str*) – Deployment environment.
- **regions** (*str*) – AWS region.

Returns True upon successful completion.

Return type bool

```
foremast.dns.destroy_dns.destroy_dns.destroy_record(client=None,  
                                                    found_record=None, record='',  
                                                    zone_id='')
```

Destroy an individual DNS record.

Parameters

- **client** (*botocore.client.Route53*) – Route 53 boto3 client.
- **found_record** (*dict*) – Route 53 record set:

```
{'Name': 'unicorn.forrest.dev.example.com.',  
 'ResourceRecords':  
  [{ 'Value':  
      'internal-unicornforrest-1777489395.us-east-1.elb.amazonaws.com'  
    }],  
 'TTL': 60,  
 'Type': 'CNAME'}
```

- **record** (*str*) – Application DNS record name. e.g.
- **zone_id** (*str*) – Route 53 Hosted Zone ID, e.g. /hostedzone/ZSVGJWJ979WQD.

Returns True upon successful completion.

Return type bool

Module contents

Submodules

foremast.dns.__main__ module

CLI entry point for creating DNS record.

Help: `python -m src.foremast.dns -h`

```
foremast.dns.__main__.main()
```

Run newer stuffs.

foremast.dns.create_dns module

Module to create dynamically generated DNS record in route53

```
class foremast.dns.create_dns.SpinnakerDns (app=None, env=None, region=None,  
                                           elb_subnet=None, prop_path=None)
```

Bases: `object`

Manipulate and create generated DNS record in Route53.

Parameters

- **app** (*str*) – application name for DNS record
- **env** (*str*) – Environment/Account for DNS record creation
- **region** (*str*) – AWS Region for DNS record
- **elb_subnet** (*str*) – Whether the DNS record is in a public or private zone
- **prop_path** (*str*) – Path to the generated property files

Returns FQDN of application

Return type *str*

create_elb_dns (*regionspecific=False*)

Create dns entries in route53.

Parameters **hasregion** (*bool*) – The DNS entry should have region on it

Returns Auto-generated DNS name for the Elastic Load Balancer.

create_failover_dns (*primary_region='us-east-1'*)

Create dns entries in route53 for multiregion failover setups

Parameters **primary_region** (*str*) – primary AWS region for failover

Returns Auto-generated DNS name.

Module contents

Package for creating DNS records in AWS

foremast.elb package

Subpackages

foremast.elb.destroy_elb package

Submodules

foremast.elb.destroy_elb.__main__ module

CLI Entry point for ELB cleanup.

`foremast.elb.destroy_elb.__main__.main()`

Destroy any ELB related Resources.

foremast.elb.destroy_elb.destroy_elb module

Destroy any ELB Resources.

`foremast.elb.destroy_elb.destroy_elb.destroy_elb` (*app='', env='dev', region='us-east-1', **_*)

Destroy ELB Resources.

Parameters

- **app** (*str*) – Spinnaker Application name.

- **env** (*str*) – Deployment environment.
- **region** (*str*) – AWS region.

Returns True upon successful completion.

Module contents

Submodules

foremast.elb.__main__ module

CLI entry point for ELB creation.

Help: `python -m src.foremast.elb -h`

`foremast.elb.__main__.main()`
Entry point for ELB creation

foremast.elb.create_elb module

Create ELBs for Spinnaker Pipelines.

class `foremast.elb.create_elb.SpinnakerELB` (*app*='', *env*='', *region*='', *prop_path*='')
Bases: `object`

Create ELBs for Spinnaker.

Parameters

- **app** (*str*) – Application name.
- **env** (*str*) – Deployment environment.
- **prop_path** (*str*) – Path to the raw.properties.json.
- **region** (*str*) – AWS Region.

add_backend_policy (*json_data*)

Attaches backend server policies to an ELB

Parameters *json_data* (`json`) – return data from ELB upsert

add_listener_policy (*json_data*)

Attaches listener policies to an ELB

Parameters *json_data* (`json`) – return data from ELB upsert

add_stickiness ()

Adds stickiness policy to created ELB

Returns

A dict of stickiness policies and ports:

```
example:
{
    80: "$policy_name"
}
```

Return type `dict`

configure_load_balancer_attributes (*json_data*)

Configure load balancer attributes such as idle timeout, connection draining, etc

Parameters *json_data* (*json*) – return data from ELB upsert

create_elb ()

Create or Update the ELB after rendering JSON data from configs. Asserts that the ELB task was successful.

make_elb_json ()

Render the JSON template with arguments.

Returns Rendered ELB template.

Return type *str*

foremast.elb.format_listeners module

Add the appropriate ELB Listeners.

`foremast.elb.format_listeners.format_cert_name` (*env=''*, *account=''*, *certificate=None*)

Format the SSL certificate name into ARN for ELB.

Parameters

- **env** (*str*) – Account environment name
- **account** (*str*) – Account number for ARN
- **certificate** (*str*) – Name of SSL certificate

Returns Fully qualified ARN for SSL certificate None: Certificate is not desired

Return type *str*

`foremast.elb.format_listeners.format_listeners` (*elb_settings=None*, *env='dev'*)

Format ELB Listeners into standard list.

Parameters

- **elb_settings** (*dict*) – ELB settings including ELB Listeners to add, e.g.:

```
# old
{
  "certificate": null,
  "i_port": 8080,
  "lb_port": 80,
  "subnet_purpose": "internal",
  "target": "HTTP:8080/health"
}

# new
{
  "ports": [
    {
      "instance": "HTTP:8080",
      "loadbalancer": "HTTP:80"
    },
    {
      "certificate": "cert_name",
      "instance": "HTTP:8443",
      "loadbalancer": "HTTPS:443"
    }
  ]
}
```

```
    }
  ],
  "subnet_purpose": "internal",
  "target": "HTTP:8080/health"
}
```

- **env** (*str*) – Environment to find the Account Number for.

Returns

ELB Listeners formatted into dicts for Spinnaker:

```
[
  {
    'externalPort': 80,
    'externalProtocol': 'HTTP',
    'internalPort': 8080,
    'internalProtocol': 'HTTP',
    'sslCertificateId': None,
    'listenerPolicies': [],
    'backendPolicies': []
  },
  ...
]
```

Return type list

`foremast.elb.format_listeners.generate_custom_cert_name` (*env*='', *account*='', *certificate*=None)

Generate a custom TLS Cert name based on a template.

Parameters

- **env** (*str*) – Account environment name
- **account** (*str*) – Account number for ARN.
- **certificate** (*str*) – Name of SSL certificate.

Returns Fully qualified ARN for SSL certificate. None: Template doesn't exist.

Return type str

foremast.elb.splay_health module

Cut Health Target up into pieces.

`foremast.elb.splay_health.splay_health` (*health_target*)

Set Health Check path, port, and protocol.

Parameters **health_target** (*str*) – The health target. ie HTTP:80

Returns A `collections.namedtuple` class with *path*, *port*, *proto*, and *target* attributes.

Return type HealthCheck

Module contents

Package for management of ELBs in Spinnaker

foremast.iam package

Subpackages

foremast.iam.destroy_iam package

Submodules

foremast.iam.destroy_iam.__main__ module

Destroy IAM command.

```
foremast.iam.destroy_iam.__main__.main()
    Destroy any IAM related Resources.
```

foremast.iam.destroy_iam.destroy_iam module

Destroy any IAM related resources.

```
foremast.iam.destroy_iam.destroy_iam.destroy_iam(app='', env='dev', **_)
    Destroy IAM Resources.
```

Parameters

- **app** (*str*) – Spinnaker Application name.
- **env** (*str*) – Deployment environment, i.e. dev, stage, prod.

Returns True upon successful completion.

Module contents

Submodules

foremast.iam.__main__ module

CLI entry point to create IAM Instance Profiles, Roles, Users, and Groups.

Help: `python -m src.foremast.iam -h`

```
foremast.iam.__main__.main()
    Command to create IAM Instance Profiles, Roles, Users, and Groups.
```

IAM Roles will retain any attached Managed Policies. Inline Policies that do not match the name *iam-project_repo_policy* will also be left untouched.

WARNING: Inline Policies named *iam-project_repo_policy* will be rewritten.

foremast.iam.construct_policy module

Construct an IAM Policy from templates.

Examples

pipeline.json:

```
{
  "services": {
    "dynamodb": [
      "another_app"
    ],
    "lambda": true,
    "s3": true
  }
}
```

`foremast.iam.construct_policy.auto_service` (*pipeline_settings*={}, *services*={})
Automatically enable service for deployment types.

Parameters

- **services** (*dict*) – Services to enable in IAM Policy.
- **pipeline_settings** (*dict*) – Settings from *pipeline.json*.

Returns Services.

Return type *dict*

`foremast.iam.construct_policy.construct_policy` (*app*='coreforrest', *env*='dev',
group='forrest', *region*='us-east-1',
pipeline_settings=None)

Assemble IAM Policy for *_app_*.

Parameters

- **app** (*str*) – Name of Spinnaker Application.
- **env** (*str*) – Environment/Account in AWS
- **group** (*str*) – A Application group/namespace
- **region** (*str*) – AWS region
- **pipeline_settings** (*dict*) – Settings from *pipeline.json*.

Returns Custom IAM Policy for *_app_*. None: When no *services* have been defined in *pipeline.json*.

Return type *json*

foremast.iam.create_iam module

Create IAM Instance Profiles, Roles, Users, and Groups.

`foremast.iam.create_iam.attach_profile_to_role` (*client*, *role_name*='forrest_unicorn_role',
profile_name='forrest_unicorn_profile')

Attach an IAM Instance Profile *_profile_name_* to Role *_role_name_*.

Parameters

- **role_name** (*str*) – Name of Role.
- **profile_name** (*str*) – Name of Instance Profile.

Returns True upon successful completion.

`foremast.iam.create_iam.create_iam_resources` (*env*='dev', *app*='', **_)
 Create the IAM Resources for the application.

Parameters

- **env** (*str*) – Deployment environment/account, i.e. dev, stage, prod.
- **app** (*str*) – Spinnaker Application name.

Returns True upon successful completion.

foremast.iam.resource_action module

Generic boto3 Resource action caller.

`foremast.iam.resource_action.resource_action` (*client*, *action*='', *log_format*='item:
 %(key)s', ***kwargs*)

Call `_action_` using `boto3_client_` with `_kwargs_`.

This is meant for `_action_` methods that will create or implicitly prove a given Resource exists. The `_log_failure_` flag is available for methods that should always succeed, but will occasionally fail due to unknown AWS issues.

Parameters

- **client** (*botocore.client.IAM*) – boto3 client object.
- **action** (*str*) – Client method to call.
- **log_format** (*str*) – Generic log message format, 'Added' or 'Found' will be prepended depending on the scenario.
- **prefix** (*str*) – Prefix word to use in successful INFO message.
- ****kwargs** – Keyword arguments to pass to `_action_` method.

Returns boto3 response.

Return type dict

Module contents

Package for managing IAM policies in AWS. Spinnaker does not directly interfact with IAM so this package mostly uses Boto3

foremast.pipeline package

Submodules

foremast.pipeline.__main__ module

CLI entry point to create Spinnaker Pipelines.

Help: `python -m src.foremast.pipeline -h`

`foremast.pipeline.__main__.main` ()
 Creates a pipeline in Spinnaker

foremast.pipeline.clean_pipelines module

Clean removed Pipelines.

`foremast.pipeline.clean_pipelines.clean_pipelines` (*app*='', *settings*=None)

Delete Pipelines for regions not defined in application.json files.

For Pipelines named **app_name** [**region**], `_region_` will need to appear in at least one application.json file. All other names are assumed unmanaged.

Parameters

- **app** (*str*) – Application name
- **settings** (*dict*) – imported configuration settings

Returns Upon successful completion.

Return type `True`

Raises

- `SpinnakerPipelineCreationFailed` – Missing application.json file from
- `create-configs`.

`foremast.pipeline.clean_pipelines.delete_pipeline` (*app*='', *pipeline_name*='')

Delete `_pipeline_name_` from `_app_`.

foremast.pipeline.construct_pipeline_block module

Construct a block section of Stages in a Spinnaker Pipeline.

`foremast.pipeline.construct_pipeline_block.check_provider_healthcheck` (*settings*,
de-
fault_provider='Discovery')

Set Provider Health Check when specified.

Returns

ProviderHealthCheck with attributes:

- `providers` (list): Providers set to use native Health Check.
- `has_healthcheck` (bool): If any native Health Checks requested.

Return type `collections.namedtuple`

`foremast.pipeline.construct_pipeline_block.construct_pipeline_block` (*env*='',
gener-
ated=None,
previ-
ous_env=None,
region='us-
east-1',
re-
gion_subnets=None,
set-
tings=None,
pipeline_data=None)

Create the Pipeline JSON from template.

This handles the common repeatable patterns in a pipeline, such as judgement, infrastructure, tagger and qe.

Parameters

- **env** (*str*) – Deploy environment name, e.g. dev, stage, prod.
- **generated** (*gogoutils.Generator*) – Gogo Application name generator.
- **previous_env** (*str*) – The previous deploy environment to use as Trigger.
- **region** (*str*) – AWS Region to deploy to.
- **settings** (*dict*) – Environment settings from configurations.
- **region_subnets** (*dict*) – Subnets for a Region, e.g. {'us-west-2': ['us-west-2a', 'us-west-2b', 'us-west-2c']}.

Returns Pipeline JSON template rendered with configurations.

Return type `dict`

foremast.pipeline.construct_pipeline_block_lambda module

Construct a block section of Stages in a Spinnaker Pipeline.

```
foremast.pipeline.construct_pipeline_block_lambda.construct_pipeline_block_lambda (env='',
gen-
er-
ated=None,
pre-
vi-
ous_env=None,
region='us-
east-
1',
re-
gion_subnets=None,
set-
tings=None,
pipeline_data=None)
```

Create the Pipeline JSON from template.

This handles the common repeatable patterns in a pipeline, such as judgement, infrastructure, tagger and qe.

Parameters

- **env** (*str*) – Deploy environment name, e.g. dev, stage, prod.
- **generated** (*gogoutils.Generator*) – Gogo Application name generator.
- **previous_env** (*str*) – The previous deploy environment to use as Trigger.
- **region** (*str*) – AWS Region to deploy to.
- **settings** (*dict*) – Environment settings from configurations.
- **region_subnets** (*dict*) – Subnets for a Region, e.g. {'us-west-2': ['us-west-2a', 'us-west-2b', 'us-west-2c']}.

Returns Pipeline JSON template rendered with configurations.

Return type `dict`

foremast.pipeline.create_pipeline module

Create Pipelines for Spinnaker.

```
class foremast.pipeline.create_pipeline.SpinnakerPipeline (app='', trigger_job='',  
                                                         prop_path='', base='',  
                                                         runway_dir='')
```

Bases: `object`

Manipulate Spinnaker Pipelines.

Parameters

- **app** (*str*) – Application name.
- **trigger_job** (*str*) – Jenkins trigger job.
- **base** (*str*) – Base image name (i.e: fedora).
- **prop_path** (*str*) – Path to the raw.properties.json.
- **runway_dir** (*str*) – Path to local runway directory.

compare_with_existing (*region='us-east-1'*)

Compare desired pipeline with existing pipelines.

Parameters **region** (*str*) – Region of desired pipeline.

Returns pipeline_id if existing, empty string if not.

Return type `str`

create_pipeline ()

Main wrapper for pipeline creation. 1. Runs `clean_pipelines` to clean up existing ones 2. determines which environments the pipeline needs 3. gets all subnets for template rendering 4. Renders all of the pipeline blocks as defined in configs 5. Runs `post_pipeline` to create pipeline

get_existing_pipelines ()

Get existing pipeline configs for specific application.

Returns Pipeline config json

Return type `str`

post_pipeline (*pipeline*)

Send Pipeline JSON to Spinnaker.

Parameters **pipeline** (*json*) – json of the pipeline to be created in Spinnaker

render_wrapper (*region='us-east-1'*)

Generate the base Pipeline wrapper.

This renders the non-repeatable stages in a pipeline, like jenkins, baking, tagging and notifications.

Parameters **region** (*str*) – AWS Region.

Returns Rendered Pipeline wrapper.

Return type `dict`

foremast.pipeline.create_pipeline_lambda module

Create Pipelines for Spinnaker.

```
class foremast.pipeline.create_pipeline_lambda.SpinnakerPipelineLambda (app='',
                                                                    trigger_job='',
                                                                    prop_path='',
                                                                    base='',
                                                                    run-
                                                                    way_dir='')
```

Bases: *foremast.pipeline.create_pipeline.SpinnakerPipeline*

Manipulate Spinnaker Pipelines.

Parameters

- **app** (*str*) – Application name.
- **trigger_job** (*str*) – Jenkins trigger job.
- **base** (*str*) – Base image name (i.e: fedora).
- **prop_path** (*str*) – Path to the raw.properties.json.

create_pipeline ()

Main wrapper for pipeline creation. 1. Runs clean_pipelines to clean up existing ones 2. determines which environments the pipeline needs 3. gets all subnets for template rendering 4. Renders all of the pipeline blocks as defined in configs 5. Runs post_pipeline to create pipeline

render_wrapper (region='us-east-1')

Generate the base Pipeline wrapper.

This renders the non-repeatable stages in a pipeline, like jenkins, baking, tagging and notifications.

Parameters **region** (*str*) – AWS Region.

Returns Rendered Pipeline wrapper.

Return type dict

foremast.pipeline.create_pipeline_onetime module

Create onetime Pipelines for Spinnaker.

These are circumventions for redployments to a specific Environment in a Region.

```
class foremast.pipeline.create_pipeline_onetime.SpinnakerPipelineOnetime (app='',
                                                                    trigger_job='',
                                                                    prop_path='',
                                                                    base='',
                                                                    one-
                                                                    time='')
```

Bases: *foremast.pipeline.create_pipeline.SpinnakerPipeline*

Manipulate Spinnaker Pipelines.

Parameters

- **app** (*str*) – Application name.
- **trigger_job** (*str*) – Jenkins trigger job.
- **base** (*str*) – Base image name (i.e: fedora).
- **prop_path** (*str*) – Path to the raw.properties.json.

- **onetime** (*str*) – Environment to build onetime pipeline for.

post_pipeline (*pipeline*)

Send Pipeline JSON to Spinnaker.

Parameters **pipeline** (*dict*, *str*) – New Pipeline to create.

foremast.pipeline.renumerate_stages module

Renumerate the Pipeline Stages.

foremast.pipeline.renumerate_stages.**renumerate_stages** (*pipeline*)

Renumber Pipeline Stage reference IDs to account for dependencies.

stage order is defined in the templates. The `refId` field dictates if a stage should be mainline or parallel to other stages.

- **master** - A mainline required stage. Other stages depend on it
- **branch** - A stage that should be ran in parallel to master stages.
- **merge** - A stage that is parallel but other stages still depend on it.

Parameters **pipeline** (*dict*) – Completed Pipeline ready for renumeration.

Returns Pipeline ready to be sent to Spinnaker.

Return type `dict`

Module contents

Package for the creation of spinnaker pipelines

foremast.s3 package

Subpackages

foremast.s3.destroy_s3 package

Submodules

foremast.s3.destroy_s3.__main__ module

Destroy any S3 Resources.

foremast.s3.destroy_s3.__main__.**main**()

Destroy any S3 Resources for a Spinnaker Application.

foremast.s3.destroy_s3.destroy_s3 module

Destroy any S3 Resources.

foremast.s3.destroy_s3.destroy_s3.**destroy_s3** (*app*='', *env*='dev', **_)

Destroy S3 Resources for `_app` in `_env`.

Parameters

- **app** (*str*) – Application name
- **env** (*str*) – Deployment environment/account name

Returns True if destroyed successfully

Return type boolean

Module contents**Submodules****foremast.s3.__main__ module**

Add application.properties to Application's S3 Bucket directory.

Help: `python -m src.foremast.s3 -h`

`foremast.s3.__main__.main()`

Create application.properties for a given application.

foremast.s3.create_archaius module

Archaius functions for deployment.

`foremast.s3.create_archaius.init_properties (env='dev', app='unnecessary', **_)`

Make sure `_application.properties_` file exists in S3.

For Applications with Archaius support, there needs to be a file where the cloud environment variable points to.

Parameters

- **env** (*str*) – Deployment environment/account, i.e. dev, stage, prod.
- **app** (*str*) – GitLab Project name.

Returns True when application.properties was found. False when application.properties needed to be created.

Module contents

Package for creating S3 bucket for each application. Used primarily for Archaius

foremast.securitygroup package**Subpackages****foremast.securitygroup.destroy_sg package****Submodules**

foremast.securitygroup.destroy_sg.__main__ module

Destroy Security Groups.

```
foremast.securitygroup.destroy_sg.__main__.main()
    Destroy any Security Group related Resources.
```

foremast.securitygroup.destroy_sg.destroy_sg module

Destroy Security Group Resources.

```
foremast.securitygroup.destroy_sg.destroy_sg.destroy_sg(app='', env='', region='',
**_)
```

Destroy Security Group.

Parameters

- **app** (*str*) – Spinnaker Application name.
- **env** (*str*) – Deployment environment.
- **region** (*str*) – Region name, e.g. us-east-1.

Returns True upon successful completion.

Module contents

Submodules

foremast.securitygroup.__main__ module

CLI entry point for security group creation

Help: `python -m src.foremast.securitygroup -h`

```
foremast.securitygroup.__main__.main()
    Entry point for creating an application specific security group
```

foremast.securitygroup.create_securitygroup module

Create Security Groups for Spinnaker Pipelines.

Security Group port specifications will be sourced from the *application.json* files for each environment.

Examples

application-master.json:

```
{
  "security_group": {
    "description": "Security Group description",
    "ingress": {
      "eureka": [
        {"start_port": 80, "end_port": 8080, "protocol": "tcp"}
      ],
    },
  },
}
```

```

        "coreforrest": [
            8080,
            8443
        ],
        "0.0.0.0/0": [
            8080
        ]
    }
}
}

```

class `foremast.securitygroup.create_securitygroup.SpinnakerSecurityGroup` (*app=None, env=None, re-gion=None, prop_path=None*)

Bases: `object`

Manipulate Spinnaker Security Groups.

Parameters

- **app** (*str*) – Application name.
- **env** (*str*) – Deployment environment.
- **prop_path** (*str*) – Path to the raw.properties.json.
- **region** (*str*) – AWS Region.

add_cidr_rules (*rules*)

Add cidr rules to security group via boto.

Parameters **rules** (*list*) – Allowed Security Group ports and protocols.

Returns Upon successful completion.

Return type `True`

Raises `SpinnakerSecurityGroupError` – boto3 call failed to add CIDR block to Security Group.

create_security_group ()

Send a POST to spinnaker to create a new security group.

Returns `True` if created successfully

Return type `boolean`

Raises `ForemastConfigurationFileError` – Missing environment configuration or misconfigured Security Group definition.

Module contents

Package for managing security groups in AWS

foremast.slacknotify package

Submodules

foremast.slacknotify.__main__ module

CLI entry point for sending Slack notifications.

Help: `python -m src.foremast.slacknotify -h`

`foremast.slacknotify.__main__.main()`
Send Slack notification to a configured channel.

foremast.slacknotify.slack_notification module

Notify Slack channel.

class `foremast.slacknotify.slack_notification.SlackNotification` (*app=None, env=None, prop_path=None*)

Bases: `object`

Post slack notification. Inform users about infrastructure changes to prod* accounts.

Parameters

- **app** (*str*) – Application name
- **env** (*str*) – Environment/account name of changed infrastructure
- **prop_path** (*str*) – Path to the rendered configuration files

notify_slack_channel ()
Post message to a defined Slack channel.

post_message ()
Send templated message to **#deployments-{env}**.
Primarily for production deployments.

Module contents

Package for posting a notification to slack

foremast.utils package

Submodules

foremast.utils.apps module

Application related utilities

`foremast.utils.apps.get_all_apps` ()
Get a list of all applications in Spinnaker.

Returns Response from Gate containing list of all apps.

Return type `requests.models.Response`

`foremast.utils.apps.get_details` (*app='groupproject', env='dev', region='us-east-1'*)
Extract details for Application.

Parameters

- **app** (*str*) – Application Name
- **env** (*str*) – Environment/account to get details from

Returns

collections.namedtuple with **_group_**, **_policy_**, **_profile_**, **_role_**, **_user_**.

foremast.utils.awslambda module

Lambda related utilities

`foremast.utils.awslambda.add_lambda_permissions` (*function*='', *statement_id*='', *action*='lambda:InvokeFunction', *principal*='', *source_arn*='', *env*='', *region*='us-east-1')

Add permission to Lambda for the event trigger.

Parameters

- **function** (*str*) – Lambda function name
- **statement_id** (*str*) – IAM policy statement (principal) id
- **action** (*str*) – Lambda action to allow
- **principal** (*str*) – AWS principal to add permissions
- **source_arn** (*str*) – ARN of the source of the event. Only needed for S3
- **env** (*str*) – Environment/account of function
- **region** (*str*) – AWS region of function

`foremast.utils.awslambda.get_lambda_alias_arn` (*app*, *account*, *region*)

Get lambda alias ARN. Assumes that account name is equal to alias name.

Parameters

- **account** (*str*) – AWS account name.
- **region** (*str*) – Region name, e.g. us-east-1
- **app** (*str*) – Lambda function name

Returns ARN for requested lambda alias

Return type *str*

`foremast.utils.awslambda.get_lambda_arn` (*app*, *account*, *region*)

Get lambda ARN.

Parameters

- **account** (*str*) – AWS account name.
- **region** (*str*) – Region name, e.g. us-east-1
- **app** (*str*) – Lambda function name

Returns ARN for requested lambda function

Return type *str*

`foremast.utils.awslambda.remove_all_lambda_permissions` (*app_name*='', *env*='',
region='us-east-1')

removes all foremast-* permissions from lambda

Parameters

- **app_name** (*str*) – Application name
- **env** (*str*) – AWS environment
- **region** (*str*) – AWS region

foremast.utils.banners module

Prints a banner.

Example:

Create Security Group

`foremast.utils.banners.banner` (*text*, *border*='=', *width*=80)
Center *text* in a banner *width* wide with *border* characters.

Parameters

- **text** (*str*) – What to write in the banner
- **border** (*str*) – Border character
- **width** (*int*) – How long the border should be

foremast.utils.credentials module

Retrieve Account Credential from Gate API.

`foremast.utils.credentials.get_env_credential` (*env*='dev')

Get Account Credential from Spinnaker for *env*.

Parameters *env* (*str*) – Environment name to find credentials for.

Returns

Complete credentials for *env*:

```
{
  'accountId': '123098123',
  'accountType': 'dev',
  'assumeRole': 'role/spinnakerManaged',
  'bastionEnabled': False,
  'challengeDestructiveActions': False,
  'cloudProvider': 'aws',
  'defaultKeyPair': 'dev_access',
  'discoveryEnabled': False,
  'eddaEnabled': False,
  'environment': 'dev',
  'front50Enabled': False,
  'name': 'dev',
  'primaryAccount': False,
  'provider': 'aws',
```

```

    'regions': [
      {
        'availabilityZones': ['us-east-1b', 'us-east-1c',
                              'us-east-1d', 'us-east-1e'],
        'deprecated': False,
        'name': 'us-east-1',
        'preferredZones':
          ['us-east-1b', 'us-east-1c', 'us-east-1d', 'us-east-1e']
      }, {
        'availabilityZones':
          ['us-west-2a', 'us-west-2b', 'us-west-2c'],
        'deprecated': False,
        'name': 'us-west-2',
        'preferredZones':
          ['us-west-2a', 'us-west-2b', 'us-west-2c']
      }
    ],
    'requiredGroupMembership': [],
    'sessionName': 'Spinnaker',
    'type': 'aws'
  }

```

Return type dict

foremast.utils.deep_chain_map module

ChainMap modification to handle nested dict objects.

class foremast.utils.deep_chain_map.**DeepChainMap**(*maps)
 Bases: collections.ChainMap

Deep lookups for collections.ChainMap objects.

When there are nested dicts, the first found, second level dict is returned instead of overlaying alternative second level dicts.

```

>>> first = {'key1': {'key1_1': 'first_one'}}
>>> second = {'key1': {'key1_1': 'second_one', 'key1_2': 'second_two'}}
>>> collections.ChainMap(first, second)['key1']
{'key1_1': 'first_one'}
>>> collections.ChainMap(second, first)['key1']
{'key1_1': 'second_one', 'key1_2': 'second_two'}

```

Deep lookup will flatten every level.

```

>>> DeepChainMap(first, second)['key1']
{'key1_1': 'first_one', 'key1_2': 'second_two'}
>>> DeepChainMap(second, first)['key1']
{'key1_1': 'second_one', 'key1_2': 'second_two'}

```

foremast.utils.dns module

Retrieve Route 53 Hosted Zone IDs.

`foremast.utils.dns.delete_existing_cname(env, zone_id, dns_name)`

Function to delete an existing CNAME record. This is used when updating to multi-region for deleting old records. The record can not just be upserted since it changes types.

Parameters

- **env** (*str*) – Deployment environment.
- **zone_id** (*str*) – Route53 zone id.
- **dns_name** (*str*) – FQDN of application’s dns entry to add/update.

`foremast.utils.dns.find_existing_record(env, zone_id, dns_name, check_key=None, check_value=None)`

Checks if a specific DNS record exists

Parameters

- **env** (*str*) – Deployment environment.
- **zone_id** (*str*) – Route53 zone id.
- **dns_name** (*str*) – FQDN of application’s dns entry to add/update.
- **check_key** (*str*) – Key to look for in record. Example: “Type”
- **check_value** (*str*) – Value to look for with check_key. Example: “CNAME”

Returns Found Record. Returns None if no record found

Return type *json*

`foremast.utils.dns.get_dns_zone_ids(env='dev', facing='internal')`

Get Route 53 Hosted Zone IDs for `_env_`.

Parameters

- **env** (*str*) – Deployment environment.
- **facing** (*str*) – Type of ELB, external or internal.

Returns Hosted Zone IDs for `_env_`. Only *PrivateZone* when `_facing_` is internal.

Return type *list*

`foremast.utils.dns.update_dns_zone_record(env, zone_id, **kwargs)`

Create a Route53 CNAME record in `_env_` zone.

Parameters

- **env** (*str*) – Deployment environment.
- **zone_id** (*str*) – Route53 zone id.

Keyword Arguments

- **dns_name** (*str*) – FQDN of application’s dns entry to add/update.
- **dns_name_aws** (*str*) – FQDN of AWS resource
- **dns_ttl** (*int*) – DNS time-to-live (ttl)

`foremast.utils.dns.update_failover_dns_record(env, zone_id, **kwargs)`

Create a Failover Route53 alias record in `_env_` zone.

Parameters

- **env** (*str*) – Deployment environment.
- **zone_id** (*str*) – Route53 zone id.

Keyword Arguments

- **dns_name** (*str*) – FQDN of application’s dns entry to add/update.
- **dns_ttl** (*int*) – DNS time-to-live (ttl)
- **elb_aws_dns** (*str*) – DNS A Record of ELB from AWS
- **elb_dns_zone_id** (*str*) – Zone ID of ELB DNS
- **failover_state** (*str*) – if the record is primary or secondary
- **primary_region** (*str*) – Primary AWS region for DNS

foremast.utils.elb module

Search for ELB DNS name.

`foremast.utils.elb.find_elb(name='', env='', region='')`

Get an application’s AWS elb dns name.

Parameters

- **name** (*str*) – ELB name
- **env** (*str*) – Environment/account of ELB
- **region** (*str*) – AWS Region

Returns elb DNS record

Return type *str*

`foremast.utils.elb.find_elb_dns_zone_id(name='', env='dev', region='us-east-1')`

Get an application’s AWS elb dns zone id.

Parameters

- **name** (*str*) – ELB name
- **env** (*str*) – Environment/account of ELB
- **region** (*str*) – AWS Region

Returns elb DNS zone ID

Return type *str*

foremast.utils.encoding module

Generate base64 encoded User Data.

`foremast.utils.encoding.generate_encoded_user_data(env='dev', region='us-east-1', app_name='', group_name='', canary=False)`

Generate base64 encoded User Data.

Parameters

- **env** (*str*) – Deployment environment, e.g. dev, stage.
- **region** (*str*) – AWS Region, e.g. us-east-1.
- **app_name** (*str*) – Application name, e.g. coreforrest.

- **group_name** (*str*) – Application group name, e.g. core.

Returns

base64 encoded User Data script.

```
#!/bin/bash      export      CLOUD_ENVIRONMENT=dev      export
CLOUD_ENVIRONMENT_C=dev      export      CLOUD_ENVIRONMENT_P=dev
export CLOUD_ENVIRONMENT_S=dev      export CLOUD_APP=coreforrest      ex-
port CLOUD_APP_GROUP=forrest      export CLOUD_STACK=forrest      export
EC2_REGION=us-east-1      export CLOUD_DOMAIN=dev.example.com      printenv |
grep 'CLOUD|EC2' | awk '$0='export "$0">> /etc/gogo/cloud_env
```

Return type *str*

foremast.utils.gate module

Dynamic Gate API interface.

It is best to use a new Class Instance for each request unless something is keeping close track of `_self.path_`. Reusing an Instance will have adverse affects as `_self.path_` will not be cleared automatically.

Examples

Typical imports:

```
from foremast.utils import Gate
from .utils import Gate
```

Responses will be a *dict* type:

```
response = Gate().credentials.dev.get()
response = Gate('credentials/dev').get()
type(response) == dict
```

The first arguments is expected to be a JSON *str* or *dict* representation and will be passed to the **json** argument of the `_requests_` call:

```
response = Gate().tasks.post({'json': 'data'})
# requests.post(json={'json': 'data'})
```

Keyword arguments besides `_json_dict_` will be passed to **data** or **params** of the `_requests_` call:

```
response = Gate().tasks.post({'json': 'data'}, custom='param')
# requests.post(json={'json': 'data'}, data={'custom': 'param'})
```

What is happening? The normal instantiation initializes the request path to an empty string:

```
a = Gate()
a.path == ''
```

Accessing attributes will append the accessed attribute to the end of the path:

```
b = Gate().applications
# __getattr__() appends 'applications' to self.path and returns self
b.path == '/applications'
```

This continues recursively because each attribute access will return a reference to the current Instance:

```
c = Gate().applications.coreforrest
# __getattr__() appends 'applications' to self.path and returns self
# __getattr__() appends 'coreforrest' to self.path and returns self
c.path == '/applications/coreforrest'
```

Once an HTTP method matches, the Instance's **verb** attribute will be set in preparation for the `_requests_` call:

```
d = Gate().applications.coreforrest.get
# __getattr__() appends 'applications' to self.path and returns self
# __getattr__() appends 'coreforrest' to self.path and returns self
# __getattr__() sets 'get' to self.verb and returns self
d.path == '/applications/coreforrest'
d.verb == 'get'
```

Finally ending with `()` will trigger the `_requests_` call. The following are all equivalent calls based on the previous examples:

```
d() == c.get()
d() == b.coreforrest.get()
d() == a.applications.coreforrest.get()
d() == Gate().applications.coreforrest.get()
# __getattr__() appends 'applications' to self.path and returns self
# __getattr__() appends 'coreforrest' to self.path and returns self
# __getattr__() sets 'get' to self.verb and returns self
# __call__() executes the request
# requests.get() is used because d.verb == 'get'
# requests.get(API_URL + '/applications/coreforrest')
```

Reusing an Instance is discouraged due to stored Instance attributes:

```
e = Gate().applications
e.coreforrest.get()
# Succeeds because e.path == '/applications/coreforest'
e.edgeforrest.get()
# Fails because e.path == '/applications/coreforrest/edgeforrest'
```

class `foremast.utils.gate.Gate` (*path*='')

Bases: `object`

Dynamic Gate API interface.

Parameters *path* (*str*) – URL path or full URL.

assemble_request_kwargs (*json_data*, *kwargs*)

Construct kwargs for final request.

Parameters

- **json_data** (*str*) – JSON content to send in request.
- **path** (*str*) – URL path to request from Gate API.

Returns

Assembled kwargs:

```
{
    'headers': {'accept': '*/*', ...},
    'data': {'extra': 'arguments', ...},
```

```
'json': {'name': 'coreforrest', ...}
}
```

Return type `dict`

`normalize_url()`

Return URL based on `_self.path_` or construct from `_API_URL_`.

foremast.utils.generate_filename module

Generate various filenames.

`foremast.utils.generate_filename.generate_packer_filename(provider, region, builder)`

Generate a filename to be used by packer.

Parameters

- **provider** (*str*) – Name of Spinnaker provider.
- **region** (*str*) – Name of provider region to use.
- **builder** (*str*) – Name of builder process type.

Returns Generated filename based on parameters.

Return type `str`

foremast.utils.get_cloudwatch_event_rule module

`foremast.utils.get_cloudwatch_event_rule.get_cloudwatch_event_rule(app_name, account, region)`

foremast.utils.get_sns_subscriptions module

`foremast.utils.get_sns_subscriptions.get_sns_subscriptions(app_name, env, region)`

Lists SNS lambda subscriptions

Returns List of Lambda subscribed SNS ARNs.

Return type `list`

foremast.utils.get_sns_topic_arn module

`foremast.utils.get_sns_topic_arn.get_sns_topic_arn(topic_name, account, region)`

Get SNS topic ARN

Parameters

- **topic_name** (*str*) – Name of the topic to lookup.
- **account** (*str*) – Environment, e.g. dev
- **region** (*str*) – Region name, e.g. us-east-1

Returns ARN for requested topic name

Return type `str`

foremast.utils.lookups module

Lookup AMI ID from a simple name.

class `foremast.utils.lookups.FileLookup` (*git_short*='', *runway_dir*='')

Bases: `object`

Retrieve files from a local filesystem or remote GitLab Server.

When `_runway_dir_` is specified, the local directory is given priority and remote Git Server will not be used.

Parameters

- **git_short** (*str*) – Short Git representation of repository, e.g. forrest/core.
- **runway_dir** (*str*) – Root of local runway directory to use instead of accessing Git.

get (*branch*='master', *filename*='')

Retrieve `_filename_` from GitLab.

Parameters

- **branch** (*str*) – Git Branch to find file.
- **filename** (*str*) – Name of file to retrieve relative to root of Git repository, or `_runway_dir_` if specified.

Returns Contents of file.

Return type `str`

json (*branch*='master', *filename*='')

Retrieve `_filename_` from GitLab.

Parameters

- **branch** (*str*) – Git Branch to find file.
- **filename** (*str*) – Name of file to retrieve.

Returns Decoded JSON.

Return type `dict`

Raises `SystemExit` – Invalid JSON provided.

local_file (*filename*)

Read the local file in `_self.runway_dir_`.

Parameters **filename** (*str*) – Name of file to retrieve relative to root of `_runway_dir_`.

Returns Contents of local file.

Return type `str`

Raises `FileNotFoundError` – Requested file missing.

remote_file (*branch*='master', *filename*='')

Read the remote file on Git Server.

Parameters

- **branch** (*str*) – Git Branch to find file.
- **filename** (*str*) – Name of file to retrieve relative to root of repository.

Returns Contents of remote file.

Return type `str`

Raises `FileNotFoundError` – Requested file missing.

`foremast.utils.lookups.ami_lookup` (*region*='us-east-1', *name*='tomcat8')

Use *_name_* to find AMI ID. If no *ami_base_url* or *gitlab_token* is provided, *_name_* is returned as the ami id

Parameters

- **region** (*str*) – AWS Region to find AMI ID.
- **name** (*str*) – Simple AMI base name to lookup.

Returns AMI ID for *_name_* in *_region_*.

Return type `str`

foremast.utils.pipelines module

Check Pipeline name to match format.

`foremast.utils.pipelines.check_managed_pipeline` (*name*='', *app_name*='')

Check a Pipeline name is a managed format **app_name** [**region**].

Parameters

- **name** (*str*) – Name of Pipeline to check.
- **app_name** (*str*) – Name of Application to find in Pipeline name.

Returns Region name from managed Pipeline name.

Return type `str`

Raises `ValueError` – Pipeline is not managed.

`foremast.utils.pipelines.get_all_pipelines` (*app*='')

Get a list of all the Pipelines in *_app_*.

Parameters **app** (*str*) – Name of Spinnaker Application.

Returns Response from Gate containing Pipelines.

Return type `requests.models.Response`

`foremast.utils.pipelines.get_pipeline_id` (*app*='', *name*='')

Get the ID for Pipeline *_name_*.

Parameters

- **app** (*str*) – Name of Spinnaker Application to search.
- **name** (*str*) – Name of Pipeline to get ID for.

Returns ID of specified Pipeline. None: Pipeline or Spinnaker Appliation not found.

Return type `str`

`foremast.utils.pipelines.normalize_pipeline_name` (*name*='')

Translate unsafe characters to underscores.

foremast.utils.properties module

Get Application properties that have been generated by *create-configs*.

```
foremast.utils.properties.get_properties(properties_file='raw.properties.json',
                                         env=None)
```

Get contents of `_properties_file_` for the `_env_`.

Parameters

- **properties_file** (*str*) – File name of *create-configs* JSON output.
- **env** (*str*) – Environment to read optionally

Returns JSON loaded Application properties for `_env_`. None: Given `_env_` was not found in *create-configs* JSON output.

Return type `dict`

foremast.utils.roles module

```
foremast.utils.roles.get_role_arn(role_name, env, region)
```

Get role ARN given role name.

Parameters

- **role_name** (*str*) – Role name to lookup
- **env** (*str*) – Environment in which to lookup
- **region** (*str*) – Region

Returns ARN if role found

foremast.utils.security_group module

Get security group id

```
foremast.utils.security_group.get_security_group_id(name='', env='', region='')
```

Get a security group ID.

Parameters

- **name** (*str*) – Security Group name to find.
- **env** (*str*) – Deployment environment to search.
- **region** (*str*) – AWS Region to search.

Returns ID of Security Group, e.g. sg-xxxx.

Return type `str`

Raises

- `AssertionError` – Call to Gate API was not successful.
- `SpinnakerSecurityGroupError` – Security Group `_name_` was not found for `_env_` in `_region_`.

foremast.utils.slack module

Post a message to slack.

`foremast.utils.slack.post_slack_message` (*message=None, channel=None, username=None, icon_emoji=None*)

Format the message and post to the appropriate slack channel.

Parameters

- **message** (*str*) – Message to post to slack
- **channel** (*str*) – Desired channel. Must start with #

foremast.utils.subnets module

Get available Subnets for specific Targets.

`foremast.utils.subnets.get_subnets` (*target='ec2', purpose='internal', env='', region=''*)

Get all availability zones for a given target.

Parameters

- **target** (*str*) – Type of subnets to look up (ec2 or elb).
- **env** (*str*) – Environment to look up.
- **region** (*str*) – AWS Region to find Subnets for.

Returns dictionary of availability zones, structured like { \$region: [\$availabilityzones] } or { \$account: \$region: [\$availabilityzone] }

Return type `az_dict`

foremast.utils.tasks module

POST a new task or check status of running task

`foremast.utils.tasks.check_task` (*taskid, timeout=120*)
wrapper for `check_task`

Parameters

- **taskid** (*str*) – the task id returned from `post_task`
- **timeout** (*int*) (*optional*) – how long to wait before failing the task

Returns polls for task status.

`foremast.utils.tasks.post_task` (*task_data*)
POST JSON to Spinnaker /tasks.

Parameters **task_data** (*str*) – the task json that needs posted.

Returns taskid.

Return type `str`

`foremast.utils.tasks.wait_for_task` (*task_data*)
Run task and check the result

Parameters **task_data** (*str*) – the task json to execute

Returns polls for task status

foremast.utils.templates module

Render Jinja2 template.

`foremast.utils.templates.get_template(template_file='; **kwargs)`
Get the Jinja2 template and renders with dict `_kwargs_`.

Parameters

- **template_file** (*str*) – name of the template file
- **kwargs** – Keywords to use for rendering the Jinja2 template.

Returns String of rendered JSON template.

`foremast.utils.templates.get_template_object(template_file='')`
Get the Jinja2 template and returns Template object

Parameters **template_file** (*str*) – name of the template file

Returns Template jinja2 object

Return type `jinja2.Template`

foremast.utils.vpc module

Get VPC ID.

`foremast.utils.vpc.get_vpc_id(account, region)`
Get vpc id.

Parameters

- **account** (*str*) – AWS account name.
- **region** (*str*) – Region name, e.g. us-east-1.

Returns ID for the requested `_account_` in `_region_`.

Return type `str`

foremast.utils.warn_user module

Overwriting the default output of warnings package.

`foremast.utils.warn_user.warning_format(message, category, *_ , **_)`
Warning format

Module contents

Package for foremast supporting utilities

Submodules

foremast.args module

Common `_argparse_` arguments.

`foremast.args.add_app(parser)`

Add an `app` flag to the `_parser_`.

`foremast.args.add_artifact_path(parser)`

Add an `artifact-path` flag to `_parser_`.

`foremast.args.add_artifact_version(parser)`

Add an `artifact-version` flag to `_parser_`.

`foremast.args.add_debug(parser)`

Add a `debug` flag to the `_parser_`.

`foremast.args.add_env(parser)`

Add an `env` flag to the `_parser_`.

`foremast.args.add_gitlab_token(parser)`

Add a `token-file` flag to the `_parser_`.

`foremast.args.add_properties(parser)`

Add a `settings` flag to the `_parser_`.

`foremast.args.add_region(parser)`

Add a `region` flag to the `_parser_`.

foremast.consts module

Load base config and export package constants.

The `foremast` configuration file is read from the following locations in descending order. First found wins.

- `./foremast/foremast.cfg`
- `~/foremast/foremast.cfg`
- `/etc/foremast/foremast.cfg`

```
; foremast.cfg
[base]
domain = example.com
envs = dev,stage,prod
regions = us-east-1,us-west-2
ami_json_url = http://s3.bucketname.com/ami_lookup.json
git_url = https://git.example.com
gate_api_url = http://gate-api.example.com:8084
templates_path = ../../foremast-templates

[credentials]
gitlab_token = 123token23423343
slack_token = 123slack3203120312

[whitelists]
asg_whitelist = application1,application2

[formats]
app = {project}{repo}
```

```
dns_elb = lb-{project}{repo}.{env}.{domain}
s3_bucket = secret-{env}-{project}

[task_timeouts]
default = 120
envs = { "dev" : { "deleteScalingPolicy": 240} }
```

`foremast.consts.extract_formats` (*config_handle*)

Get application formats.

Parameters `config_handle` (*configparser.ConfigParser*) – Instance of configurations.

Returns `str` when *key* exists, otherwise *default* object. dict: of formats in {*format_type*: *format_pattern*}. See (`gogoutils.Formats`) for available options.

Return type `object`

`foremast.consts.find_config` ()

Look for `foremast.cfg` in `config_locations`.

Raises `SystemExit` – No configuration file found.

Returns found configuration file

Return type `ConfigParser`

`foremast.consts.load_dynamic_config` (*configurations*, *config_dir*=`'/home/docs/checkouts/readthedocs.org/user_builds/fo`

Load and parse dynamic config

`foremast.consts.validate_key_values` (*config_handle*, *section*, *key*, *default*=`None`)

Warn when *key* is missing from configuration *section*.

Parameters

- **config_handle** (*configparser.ConfigParser*) – Instance of configurations.
- **section** (*str*) – Name of configuration section to retrieve.
- **key** (*str*) – Configuration key to look up.
- **default** (*object*) – Default object to use when *key* is not found.

Returns `str` when *key* exists, otherwise *default* object.

Return type `object`

foremast.destroyer module

Complete Application destroyer script.

`foremast.destroyer.main` ()

Attempt to fully destroy AWS Resources for a Spinnaker Application.

foremast.exceptions module

Foremast and Spinnaker related custom exceptions.

exception `foremast.exceptions.ForemastConfigurationFileError`

Bases: `foremast.exceptions.ForemastError`

Foremast configuration file misconfigured.

exception `foremast.exceptions.ForemastError`

Bases: `Exception`

Foremast related error.

exception `foremast.exceptions.ForemastTemplateNotFound`

Bases: `Exception`

Foremast Template was not found.

exception `foremast.exceptions.InvalidEventConfiguration`

Bases: `foremast.exceptions.ForemastError`

Invalid AWS Lambda event configuration.

exception `foremast.exceptions.LambdaAliasDoesNotExist`

Bases: `foremast.exceptions.ForemastError`

Lambda function was not found.

exception `foremast.exceptions.LambdaFunctionDoesNotExist`

Bases: `foremast.exceptions.ForemastError`

Lambda function was not found.

exception `foremast.exceptions.PrimaryDNSRecordNotFound`

Bases: `foremast.exceptions.ForemastError`

Required Primary DNS record does not exist

exception `foremast.exceptions.RequiredKeyNotFound`

Bases: `foremast.exceptions.ForemastError`

Required key in json config not found

exception `foremast.exceptions.S3ArtifactNotFound`

Bases: `foremast.exceptions.ForemastError`

Could not find Artifact to upload to S3

exception `foremast.exceptions.S3SharedBucketNotFound`

Bases: `foremast.exceptions.ForemastError`

Shared S3 Bucket does not exist

exception `foremast.exceptions.SNSSubscriptionDoesNotExist`

Bases: `foremast.exceptions.ForemastError`

SNS Subscriptions does not exist.

exception `foremast.exceptions.SNSTopicNotFound`

Bases: `foremast.exceptions.ForemastError`

SNS Topic was not found.

exception `foremast.exceptions.SpinnakerAppNotFound`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker app not found error.

exception `foremast.exceptions.SpinnakerApplicationListError`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker application list error.

exception `foremast.exceptions.SpinnakerDnsCreationFailed`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker DNS creation error.

exception `foremast.exceptions.SpinnakerElbNotFound`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker Elb not found.

exception `foremast.exceptions.SpinnakerError`

Bases: `foremast.exceptions.ForemastError`

Spinnaker related error.

exception `foremast.exceptions.SpinnakerPipelineCreationFailed`

Bases: `foremast.exceptions.SpinnakerError`

Could not create Spinnaker Pipeline.

exception `foremast.exceptions.SpinnakerPipelineDeletionFailed`

Bases: `foremast.exceptions.SpinnakerError`

Could not delete Spinnaker Pipeline.

exception `foremast.exceptions.SpinnakerSecurityGroupCreationFailed`

Bases: `foremast.exceptions.SpinnakerError`

Could not create Security Group.

exception `foremast.exceptions.SpinnakerSecurityGroupError`

Bases: `foremast.exceptions.SpinnakerError`

Could not create Security Group.

exception `foremast.exceptions.SpinnakerSubnetError` (*env=''*, *region=''*)

Bases: `foremast.exceptions.SpinnakerError`

Unavailable environment or region.

exception `foremast.exceptions.SpinnakerTaskError` (*task_state*)

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker Task did not finish properly.

exception `foremast.exceptions.SpinnakerTimeout`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker Timeout error.

exception `foremast.exceptions.SpinnakerVPCIDNotFound`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker did not find the VPC ID.

exception `foremast.exceptions.SpinnakerVPCNotFound`

Bases: `foremast.exceptions.SpinnakerError`

Spinnaker did not find a VPC.

foremast.runner module

A runner for all of the spinnaker pipe modules.

Read environment variables from Jenkins:

- EMAIL
- ENV
- GIT_REPO
- PROJECT
- REGION

Then run specific prepare jobs.

class `foremast.runner.ForemastRunner`

Bases: `object`

Wrap each pipes module in a way that is easy to invoke.

cleanup ()

Clean up generated files.

create_app ()

Create the spinnaker application.

create_archaius ()

Create S3 bucket for Archaius.

create_autoscaling_policy ()

Create Scaling Policy for app in environment

create_awslambda ()

Create security groups as defined in the configs.

create_dns ()

Create DNS for the defined app and environment.

create_elb ()

Create the ELB for the defined environment.

create_iam ()

Create IAM resources.

create_pipeline (*onetime=None*)

Create the spinnaker pipeline(s).

create_s3app ()

Create S3 infra for s3 applications

create_secgroups ()

Create security groups as defined in the configs.

deploy_s3app ()

Deploys artifacts contents to S3 bucket

promote_s3app ()

promotes S3 deployment to LATEST

slack_notify ()

Send out a slack notification.

write_configs ()

Generate the configurations needed for pipes.

`foremast.runner.create_scaling_policy` ()

Create Auto Scaling Policy for an Auto Scaling Group.

`foremast.runner.debug_flag()`
Set logging level for entry points.

`foremast.runner.deploy_s3app()`
Entry point for application setup and s3 deployments

`foremast.runner.prepare_app_pipeline()`
Entry point for application setup and initial pipeline in Spinnaker.

`foremast.runner.prepare_infrastructure()`
Entry point for preparing the infrastructure in a specific env.

`foremast.runner.prepare_onetime_pipeline()`
Entry point for single use pipeline setup in the defined app.

`foremast.runner.promote_s3app()`
Entry point for application setup and s3 promotions

`foremast.runner.rebuild_pipelines(*args)`
Entry point for rebuilding pipelines.
Use to rebuild all pipelines or a specific group.

Module contents

Tools for creating infrastructure and Spinnaker Applications.

Foremast is a Spinnaker pipeline and infrastructure configuration and templating tool. Just create a couple JSON configuration files and then manually creating Spinnaker pipelines becomes a thing of the past.

Why Foremast?

- No manual creation of pipelines in the Spinnaker UI
- Reproducible and versioned Spinnaker pipelines
- Standardized pipelines with flexibility for application specific needs

With Foremast, Developers create a couple simple JSON configs per application. These configs provide details on the pipeline and infrastructure specific to the application's needs. Foremast takes those configs, renders some Jinja2 templates, and then acts as a client for the Spinnaker Gate API. Foremast comes with generic templates for creating a simple pipeline but it can also point to external templates for custom pipelines that fit any workflow.

Foremast Features

- Dynamically generate Spinnaker pipelines based on JSON configs
- Customizable pipelines through external Jinja2 Templates, see [Foremast templates](#) for examples
- Dynamically generate AWS infrastructure based on pipeline configs
- Set up resources not defined in Spinnaker, such as S3 buckets and IAM roles
- Support for AWS Lambda pipelines

Getting Started

Take a look at [quick start guide](#) for a quick introduction on how to use Foremast.

We also have a blog post to help you get started: [Automate Spinnaker Pipeline Creation](#)

Documentation

All the documentation can be viewed on [Read the Docs](#). You can find all configuration options, code information, and better examples there.

Development

See the [contribution guide](#) for information on code style, contributing, and testing.

Getting Help

For questions, support, or friendly conversation you can find us on [Gitter](#).

More Details

Installing

Installing the package will provide CLI commands for convenience.

```
virtualenv -p python3 venv
source venv/bin/activate
pip install foremast
```

Entry Points

Foremast has a few easy to use CLI endpoints.

- `foremast-pipeline` - Creates an application and pipeline Spinnaker
- `foremast-infrastructure` - Sets up AWS infrastructure like s3, iam, elb, and security groups
- `foremast-pipeline-onetime` - Generates a pipeline for deploying to one specific account
- `foremast-scaling-policy` - Creates and attaches a scaling policy to an application server group.
- `foremast-pipeline-rebuild` - rebuild pipelines after changes have been made

You can run any of these entries points from the command line. They rely on environment variables and are ideal for running in a Jenkins job

```
PROJECT=forrest GIT_REPO=core RUNWAY_DIR=path/to/pipeline_configs foremast-pipeline
```

Foremast Configuration

A file at `{pwd}/.foremast/foremast.cfg`, `~/.foremast/foremast.cfg`, or `/etc/foremast/foremast.cfg` needs to exist in order to run foremast.

```
[base]
domain = example.com
envs = dev,stage,prod
regions = us-east-1
gate_api_url = http://gate.example.com:8084
```

Runway Configuration Files

To begin using Foremast, you must have a few JSON configuration files defined for each application

pipeline.json

This file will be needed for each application. Foremast has a lot of defaults in place for `pipeline.json`, take a look at the [pipeline.json docs](#) for all options.

Minimum

```
{
  "deployment": "spinnaker"
}
```

Example Deployment Environments Override

Custom deployment environment order and selection can be provided in the `env` key. When missing, the default provided is `{"env": ["stage", "prod"]}`. Here, the order matters and Pipeline will be generated in the given order.

```
{
  "deployment": "spinnaker",
  "env": [
    "prod"
  ]
}
```

application-master-{env}.json

Each deployment environment specified in the `pipeline.json` file will need an accompanying `application-master-{env}.json` file in the same directory.

The `application-master-{env}` files have a lot of exposed values with sane defaults. Please take a look at the [application.json docs](#) for all options.

application-master-{env}.json example

```
{
  "security_group": {
    "description": "something useful",
    "elb_extras": ["sg_offices"],
    "ingress": {
    },
    "egress": "0.0.0.0/0"
  },
  "app": {
    "instance_type": "t2.small",
    "app_description": "Edge Forrest Demo application",
```

```
    "instance_profile": "forrest_edge_profile"
  },
  "elb": {
    "subnet_purpose": "internal",
    "target": "TCP:8080",
    "ports": [
      {"loadbalancer": "HTTP:80", "instance": "HTTP:8080"}
    ]
  },
  "asg": {
    "subnet_purpose": "internal",
    "min_inst": 1,
    "max_inst": 1,
    "scaling_policy": {
      "metric": "CPUUtilization",
      "threshold": 90,
      "period_minutes": 10,
      "statistic": "Average"
    }
  },
  "regions": ["us-east-1"],
  "dns": {
    "ttl": 120
  }
}
```

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

Python Module Index

f

foremast, 91
foremast.app, 45
foremast.app.__main__, 44
foremast.app.create_app, 44
foremast.args, 86
foremast.autoscaling_policy, 46
foremast.autoscaling_policy.__main__, 45
foremast.autoscaling_policy.create_policy, 45
foremast.awslambda, 54
foremast.awslambda.__main__, 53
foremast.awslambda.api_gateway_event, 48
foremast.awslambda.api_gateway_event.__main__, 47
foremast.awslambda.api_gateway_event.api_gateway_event, 47
foremast.awslambda.awslambda, 53
foremast.awslambda.awslambdaevent, 54
foremast.awslambda.cloudwatch_event, 49
foremast.awslambda.cloudwatch_event.__main__, 49
foremast.awslambda.cloudwatch_event.cloudwatch_event, 49
foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event, 49
foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.__main__, 48
foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.prepare_configs, 48
foremast.awslambda.cloudwatch_log_event, 50
foremast.awslambda.cloudwatch_log_event.__main__, 50
foremast.awslambda.cloudwatch_log_event.cloudwatch_log_event, 50
foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event, 50
foremast.aws_lambda.cloudwatch_log_event.destroy_cloudwatch_log_event, 50
foremast.aws_lambda.cloudwatch_log_event.destroy_cloudwatch_log_event.__main__, 49
foremast.aws_lambda.cloudwatch_log_event.destroy_cloudwatch_log_event.prepare_configs, 50
foremast.aws_lambda.s3_event, 52
foremast.aws_lambda.s3_event.__main__, 51
foremast.aws_lambda.s3_event.destroy_s3_event, 51
foremast.aws_lambda.s3_event.destroy_s3_event.__main__, 51
foremast.aws_lambda.s3_event.destroy_s3_event.destroy_s3_event, 51
foremast.aws_lambda.s3_event.s3_event, 51
foremast.aws_lambda.sns_event, 53
foremast.aws_lambda.sns_event.__main__, 52
foremast.aws_lambda.sns_event.destroy_sns_event, 52
foremast.aws_lambda.sns_event.destroy_sns_event.__main__, 52
foremast.aws_lambda.sns_event.destroy_sns_event.destroy_sns_event, 52
foremast.aws_lambda.sns_event.sns_event, 53
foremast.configs, 55
foremast.configs.__main__, 54
foremast.configs.outputs, 54
foremast.configs.prepare_configs, 55
foremast.consts, 86
foremast.destroyer, 87
foremast.dns, 57
foremast.dns.__main__, 56
foremast.dns.create_dns, 56
foremast.dns.destroy_dns, 56
foremast.dns.destroy_dns.__main__, 55
foremast.dns.destroy_dns.destroy_dns, 55

- foremast.elb, 60
- foremast.elb.__main__, 58
- foremast.elb.create_elb, 58
- foremast.elb.destroy_elb, 58
- foremast.elb.destroy_elb.__main__, 57
- foremast.elb.destroy_elb.destroy_elb, 57
- foremast.elb.format_listeners, 59
- foremast.elb.splay_health, 60
- foremast.exceptions, 87
- foremast.iam, 63
- foremast.iam.__main__, 61
- foremast.iam.construct_policy, 61
- foremast.iam.create_iam, 62
- foremast.iam.destroy_iam, 61
- foremast.iam.destroy_iam.__main__, 61
- foremast.iam.destroy_iam.destroy_iam, 61
- foremast.iam.resource_action, 63
- foremast.pipeline, 68
- foremast.pipeline.__main__, 63
- foremast.pipeline.clean_pipelines, 64
- foremast.pipeline.construct_pipeline_block, 64
- foremast.pipeline.construct_pipeline_block_lambda, 65
- foremast.pipeline.create_pipeline, 66
- foremast.pipeline.create_pipeline_lambda, 66
- foremast.pipeline.create_pipeline_onetime, 67
- foremast.pipeline.renumerate_stages, 68
- foremast.runner, 89
- foremast.s3, 69
- foremast.s3.__main__, 69
- foremast.s3.create_archaius, 69
- foremast.s3.destroy_s3, 69
- foremast.s3.destroy_s3.__main__, 68
- foremast.s3.destroy_s3.destroy_s3, 68
- foremast.securitygroup, 71
- foremast.securitygroup.__main__, 70
- foremast.securitygroup.create_securitygroup, 70
- foremast.securitygroup.destroy_sg, 70
- foremast.securitygroup.destroy_sg.__main__, 70
- foremast.securitygroup.destroy_sg.destroy_sg, 70
- foremast.slacknotify, 72
- foremast.slacknotify.__main__, 72
- foremast.slacknotify.slack_notification, 72
- foremast.utils, 85
- foremast.utils.apps, 72
- foremast.utils.awslambda, 73
- foremast.utils.banners, 74
- foremast.utils.credentials, 74
- foremast.utils.deep_chain_map, 75
- foremast.utils.dns, 75
- foremast.utils.elb, 77
- foremast.utils.encoding, 77
- foremast.utils.gate, 78
- foremast.utils.generate_filename, 80
- foremast.utils.get_cloudwatch_event_rule, 80
- foremast.utils.get_sns_subscriptions, 80
- foremast.utils.get_sns_topic_arn, 80
- foremast.utils.lookups, 81
- foremast.utils.pipelines, 82
- foremast.utils.properties, 83
- foremast.utils.roles, 83
- foremast.utils.security_group, 83
- foremast.utils.slack, 84
- foremast.utils.subnets, 84
- foremast.utils.tasks, 84
- foremast.utils.templates, 85
- foremast.utils.vpc, 85
- foremast.utils.warn_user, 85

A

- add_app() (in module `foremast.args`), 86
 - add_artifact_path() (in module `foremast.args`), 86
 - add_artifact_version() (in module `foremast.args`), 86
 - add_backend_policy() (foremast.elb.create_elb.SpinnakerELB method), 58
 - add_cidr_rules() (foremast.securitygroup.create_securitygroup.SpinnakerSecurityGroup method), 71
 - add_debug() (in module `foremast.args`), 86
 - add_env() (in module `foremast.args`), 86
 - add_gitlab_token() (in module `foremast.args`), 86
 - add_integration_response() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 - add_lambda_integration() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 - add_lambda_permissions() (in module `foremast.utils.awslambda`), 73
 - add_listener_policy() (foremast.elb.create_elb.SpinnakerELB method), 58
 - add_permission() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 - add_properties() (in module `foremast.args`), 86
 - add_region() (in module `foremast.args`), 86
 - add_stickiness() (foremast.elb.create_elb.SpinnakerELB method), 58
 - ami_lookup() (in module `foremast.utils.lookups`), 82
 - APIGateway (class in `foremast.awslambda.api_gateway_event.api_gateway_event`), 47
 - appinfo (foremast.app.create_app.SpinnakerApp attribute), 45
 - appname (foremast.app.create_app.SpinnakerApp attribute), 45
 - assemble_request_kwargs() (foremast.utils.gate.Gate method), 79
 - attach_method() (foremast.awslambda.api_gateway_event.api_gateway_event method), 47
 - attach_profile_to_role() (in module `foremast.iam.create_iam`), 62
 - auto_service() (in module `foremast.iam.construct_policy`), 62
 - AutoScalingPolicy (class in `foremast.autoscaling_policy.create_policy`), 45
- B**
- banner() (in module `foremast.utils.banners`), 74
- C**
- check_managed_pipeline() (in module `foremast.utils.pipelines`), 82
 - check_provider_healthcheck() (in module `foremast.pipeline.construct_pipeline_block`), 64
 - check_task() (in module `foremast.utils.tasks`), 84
 - clean_pipelines() (in module `foremast.pipeline.clean_pipelines`), 64
 - cleanup() (foremast.runner.ForemastRunner method), 90
 - compare_with_existing() (foremast.pipeline.create_pipeline.SpinnakerPipeline method), 66
 - configure_load_balancer_attributes() (foremast.elb.create_elb.SpinnakerELB method), 58
 - construct_pipeline_block() (in module `foremast.pipeline.construct_pipeline_block`), 64
 - construct_pipeline_block_lambda() (in module `foremast.pipeline.construct_pipeline_block_lambda`), 65
 - construct_policy() (in module `foremast.iam.construct_policy`), 62
 - convert_ini() (in module `foremast.configs.outputs`), 54
 - create_alias() (foremast.awslambda.awslambda.LambdaFunction method), 53

create_api() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 create_api_deployment() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 create_api_key() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 47
 create_app() (foremast.app.create_app.SpinnakerApp method), 45
 create_app() (foremast.runner.ForemastRunner method), 90
 create_archaius() (foremast.runner.ForemastRunner method), 90
 create_autoscaling_policy() (foremast.runner.ForemastRunner method), 90
 create_awslambda() (foremast.runner.ForemastRunner method), 90
 create_cloudwatch_event() (in module foremast.awslambda.cloudwatch_event.cloudwatch_event), 49
 create_cloudwatch_log_event() (in module foremast.awslambda.cloudwatch_log_event.cloudwatch_log_event), 50
 create_dns() (foremast.runner.ForemastRunner method), 90
 create_elb() (foremast.elb.create_elb.SpinnakerELB method), 59
 create_elb() (foremast.runner.ForemastRunner method), 90
 create_elb_dns() (foremast.dns.create_dns.SpinnakerDns method), 57
 create_failover_dns() (foremast.dns.create_dns.SpinnakerDns method), 57
 create_function() (foremast.awslambda.awslambda.LambdaFunction method), 53
 create_iam() (foremast.runner.ForemastRunner method), 90
 create_iam_resources() (in module foremast.iam.create_iam), 62
 create_lambda_events() (foremast.awslambda.awslambdaevent.LambdaEvent method), 54
 create_lambda_function() (foremast.awslambda.awslambda.LambdaFunction method), 53
 create_pipeline() (foremast.pipeline.create_pipeline.SpinnakerPipeline method), 66
 create_pipeline() (foremast.pipeline.create_pipeline_lambda.SpinnakerPipelineLambda method), 67
 create_pipeline() (foremast.runner.ForemastRunner method), 90
 create_policy() (foremast.autoscaling_policy.create_policy.AutoScalingPolicy method), 46
 create_policy() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 46
 create_s3_event() (in module foremast.awslambda.s3_event.s3_event), 51
 create_s3app() (foremast.runner.ForemastRunner method), 90
 create_scaling_policy() (in module foremast.runner), 90
 create_secgroups() (foremast.runner.ForemastRunner method), 90
 create_security_group() (foremast.securitygroup.create_securitygroup.SpinnakerSecurityGroup method), 71
 create_sns_event() (in module foremast.awslambda.sns_event.sns_event), 53

D

debug_flag() (in module foremast.runner), 90
 DeepChainMap (class in foremast.utils.deep_chain_map), 75
 delete_existing_cname() (in module foremast.utils.dns), 75
 delete_existing_policy() (foremast.autoscaling_policy.create_policy.AutoScalingPolicy method), 46
 delete_pipeline() (in module foremast.pipeline.clean_pipelines), 64
 deploy_s3app() (foremast.runner.ForemastRunner method), 90
 deploy_s3app() (in module foremast.runner), 91
 destroy_cloudwatch_event() (in module foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event), 48
 destroy_cloudwatch_log_event() (in module foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event), 50
 destroy_dns() (in module foremast.dns.destroy_dns.destroy_dns), 55
 destroy_elb() (in module foremast.elb.destroy_elb.destroy_elb), 57
 destroy_iam() (in module foremast.iam.destroy_iam.destroy_iam), 61
 destroy_record() (in module foremast.dns.destroy_dns.destroy_dns), 56
 destroy_s3() (in module foremast.s3.destroy_s3.destroy_s3), 68
 destroy_s3_event() (in module foremast.awslambda.s3_event.destroy_s3_event.destroy_s3_event), 51
 destroy_sg() (in module foremast.securitygroup.destroy_sg.destroy_sg), 71

- 70
- destroy_sns_event() (in module foremast.awslambda.sns_event.destroy_sns_event (module), 52)
- E**
- extract_formats() (in module foremast.consts), 87
- F**
- FileLookup (class in foremast.utils.lookups), 81
- find_api_id() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 48
- find_config() (in module foremast.consts), 87
- find_elb() (in module foremast.utils.elb), 77
- find_elb_dns_zone_id() (in module foremast.utils.elb), 77
- find_existing_record() (in module foremast.utils.dns), 76
- find_resource_id() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 48
- foremast (module), 91
- foremast.app (module), 45
- foremast.app.__main__ (module), 44
- foremast.app.create_app (module), 44
- foremast.args (module), 86
- foremast.autoscaling_policy (module), 46
- foremast.autoscaling_policy.__main__ (module), 45
- foremast.autoscaling_policy.create_policy (module), 45
- foremast.awslambda (module), 54
- foremast.awslambda.__main__ (module), 53
- foremast.awslambda.api_gateway_event (module), 48
- foremast.awslambda.api_gateway_event.__main__ (module), 47
- foremast.awslambda.api_gateway_event.api_gateway_event (module), 47
- foremast.awslambda.awslambda (module), 53
- foremast.awslambda.awslambdaevent (module), 54
- foremast.awslambda.cloudwatch_event (module), 49
- foremast.awslambda.cloudwatch_event.__main__ (module), 49
- foremast.awslambda.cloudwatch_event.cloudwatch_event (module), 49
- foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event (module), 49
- foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.__main__ (module), 48
- foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.destroy_cloudwatch_event (module), 48
- foremast.awslambda.cloudwatch_log_event (module), 50
- foremast.awslambda.cloudwatch_log_event.__main__ (module), 50
- foremast.awslambda.cloudwatch_log_event.cloudwatch_log_event (module), 50
- foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event (module), 50
- foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.__main__ (module), 49
- foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.destroy_cloudwatch_log_event (module), 50
- foremast.awslambda.s3_event (module), 52
- foremast.awslambda.s3_event.__main__ (module), 51
- foremast.awslambda.s3_event.destroy_s3_event (module), 51
- foremast.awslambda.s3_event.destroy_s3_event.__main__ (module), 51
- foremast.awslambda.s3_event.s3_event (module), 51
- foremast.awslambda.sns_event (module), 53
- foremast.awslambda.sns_event.__main__ (module), 52
- foremast.awslambda.sns_event.destroy_sns_event (module), 52
- foremast.awslambda.sns_event.destroy_sns_event.__main__ (module), 52
- foremast.awslambda.sns_event.destroy_sns_event.destroy_sns_event (module), 52
- foremast.awslambda.sns_event.sns_event (module), 53
- foremast.configs (module), 55
- foremast.configs.__main__ (module), 54
- foremast.configs.outputs (module), 54
- foremast.configs.prepare_configs (module), 55
- foremast.consts (module), 86
- foremast.destroyer (module), 87
- foremast.dns (module), 57
- foremast.dns.__main__ (module), 56
- foremast.dns.create_dns (module), 56
- foremast.dns.destroy_dns (module), 56
- foremast.dns.destroy_dns.__main__ (module), 55
- foremast.dns.destroy_dns.destroy_dns (module), 55
- foremast.elb (module), 60
- foremast.elb.__main__ (module), 58
- foremast.elb.create_elb (module), 58
- foremast.elb.destroy_elb (module), 58
- foremast.elb.destroy_elb.__main__ (module), 57
- foremast.elb.destroy_elb.destroy_elb (module), 57
- foremast.elb.format_listeners (module), 59
- foremast.elb.splay_health (module), 60
- foremast.exceptions (module), 87
- foremast.iam (module), 63
- foremast.iam.__main__ (module), 61
- foremast.iam.construct_policy (module), 61
- foremast.iam.create_iam (module), 62
- foremast.iam.destroy_iam (module), 61
- foremast.iam.destroy_iam.__main__ (module), 61
- foremast.iam.destroy_iam.destroy_iam (module), 61
- foremast.iam.resource_action (module), 63
- foremast.pipeline (module), 68
- foremast.pipeline.__main__ (module), 63
- foremast.pipeline.clean_pipelines (module), 64

- foremast.pipeline.construct_pipeline_block (module), 64
 - foremast.pipeline.construct_pipeline_block_lambda (module), 65
 - foremast.pipeline.create_pipeline (module), 66
 - foremast.pipeline.create_pipeline_lambda (module), 66
 - foremast.pipeline.create_pipeline_onetime (module), 67
 - foremast.pipeline.renumerate_stages (module), 68
 - foremast.runner (module), 89
 - foremast.s3 (module), 69
 - foremast.s3.__main__ (module), 69
 - foremast.s3.create_archaius (module), 69
 - foremast.s3.destroy_s3 (module), 69
 - foremast.s3.destroy_s3.__main__ (module), 68
 - foremast.s3.destroy_s3.destroy_s3 (module), 68
 - foremast.securitygroup (module), 71
 - foremast.securitygroup.__main__ (module), 70
 - foremast.securitygroup.create_securitygroup (module), 70
 - foremast.securitygroup.destroy_sg (module), 70
 - foremast.securitygroup.destroy_sg.__main__ (module), 70
 - foremast.securitygroup.destroy_sg.destroy_sg (module), 70
 - foremast.slacknotify (module), 72
 - foremast.slacknotify.__main__ (module), 72
 - foremast.slacknotify.slack_notification (module), 72
 - foremast.utils (module), 85
 - foremast.utils.apps (module), 72
 - foremast.utils.awslambda (module), 73
 - foremast.utils.banners (module), 74
 - foremast.utils.credentials (module), 74
 - foremast.utils.deep_chain_map (module), 75
 - foremast.utils.dns (module), 75
 - foremast.utils.elb (module), 77
 - foremast.utils.encoding (module), 77
 - foremast.utils.gate (module), 78
 - foremast.utils.generate_filename (module), 80
 - foremast.utils.get_cloudwatch_event_rule (module), 80
 - foremast.utils.get_sns_subscriptions (module), 80
 - foremast.utils.get_sns_topic_arn (module), 80
 - foremast.utils.lookups (module), 81
 - foremast.utils.pipelines (module), 82
 - foremast.utils.properties (module), 83
 - foremast.utils.roles (module), 83
 - foremast.utils.security_group (module), 83
 - foremast.utils.slack (module), 84
 - foremast.utils.subnets (module), 84
 - foremast.utils.tasks (module), 84
 - foremast.utils.templates (module), 85
 - foremast.utils.vpc (module), 85
 - foremast.utils.warn_user (module), 85
 - ForemastConfigurationFileError, 87
 - ForemastError, 87
 - ForemastRunner (class in foremast.runner), 90
 - ForemastTemplateNotFound, 88
 - format_cert_name() (in module foremast.elb.format_listeners), 59
 - format_listeners() (in module foremast.elb.format_listeners), 59
- ## G
- Gate (class in foremast.utils.gate), 79
 - generate_custom_cert_name() (in module foremast.elb.format_listeners), 60
 - generate_encoded_user_data() (in module foremast.utils.encoding), 77
 - generate_packer_filename() (in module foremast.utils.generate_filename), 80
 - generate_uris() (foremast.awslambda.api_gateway_event.api_gateway_event method), 48
 - get() (foremast.utils.lookups.FileLookup method), 81
 - get_accounts() (foremast.app.create_app.SpinnakerApp method), 45
 - get_all_apps() (in module foremast.utils.apps), 72
 - get_all_existing() (foremast.autoscaling_policy.create_policy.AutoScalingPolicy method), 46
 - get_all_pipelines() (in module foremast.utils.pipelines), 82
 - get_cloudwatch_event_rule() (in module foremast.utils.get_cloudwatch_event_rule), 80
 - get_details() (in module foremast.utils.apps), 72
 - get_dns_zone_ids() (in module foremast.utils.dns), 76
 - get_env_credential() (in module foremast.utils.credentials), 74
 - get_existing_pipelines() (foremast.pipeline.create_pipeline.SpinnakerPipeline method), 66
 - get_lambda_alias_arn() (in module foremast.utils.awslambda), 73
 - get_lambda_arn() (in module foremast.utils.awslambda), 73
 - get_pipeline_id() (in module foremast.utils.pipelines), 82
 - get_properties() (in module foremast.utils.properties), 83
 - get_role_arn() (in module foremast.utils.roles), 83
 - get_security_group_id() (in module foremast.utils.security_group), 83
 - get_server_group() (foremast.autoscaling_policy.create_policy.AutoScalingPolicy method), 46
 - get_sns_subscriptions() (in module foremast.utils.get_sns_subscriptions), 80
 - get_sns_topic_arn() (in module foremast.utils.get_sns_topic_arn), 80
 - get_subnets() (in module foremast.utils.subnets), 84
 - get_template() (in module foremast.utils.templates), 85
 - get_template_object() (in module foremast.utils.templates), 85

get_vpc_id() (in module foremast.utils.vpc), 85

I

init_properties() (in module foremast.s3.create_archaius), 69

InvalidEventConfiguration, 88

J

json() (foremast.utils.lookups.FileLookup method), 81

L

LambdaAliasDoesNotExist, 88

LambdaEvent (class in foremast.awslambda.awslambdaevent), 54

LambdaFunction (class in foremast.awslambda.awslambda), 53

LambdaFunctionDoesNotExist, 88

load_dynamic_config() (in module foremast.consts), 87

local_file() (foremast.utils.lookups.FileLookup method), 81

log (foremast.autoscaling_policy.create_policy.AutoScalingPolicy attribute), 46

M

main() (in module foremast.app.__main__), 44

main() (in module foremast.autoscaling_policy.__main__), 45

main() (in module foremast.awslambda.__main__), 53

main() (in module foremast.awslambda.api_gateway_event.__main__), 47

main() (in module foremast.awslambda.cloudwatch_event.__main__), 49

main() (in module foremast.awslambda.cloudwatch_event.destroy_cloudwatch_event.__main__), 48

main() (in module foremast.awslambda.cloudwatch_log_event.__main__), 50

main() (in module foremast.awslambda.cloudwatch_log_event.destroy_cloudwatch_log_event.__main__), 49

main() (in module foremast.awslambda.s3_event.__main__), 51

main() (in module foremast.awslambda.s3_event.destroy_s3_event.__main__), 51

main() (in module foremast.awslambda.sns_event.__main__), 52

main() (in module foremast.awslambda.sns_event.destroy_sns_event.__main__), 52

main() (in module foremast.configs.__main__), 54

main() (in module foremast.destroyer), 87

main() (in module foremast.dns.__main__), 56

main() (in module foremast.dns.destroy_dns.__main__), 55

main() (in module foremast.elb.__main__), 58

main() (in module foremast.elb.destroy_elb.__main__), 57

main() (in module foremast.iam.__main__), 61

main() (in module foremast.iam.destroy_iam.__main__), 61

main() (in module foremast.pipeline.__main__), 63

main() (in module foremast.s3.__main__), 69

main() (in module foremast.s3.destroy_s3.__main__), 68

main() (in module foremast.securitygroup.__main__), 70

main() (in module foremast.securitygroup.destroy_sg.__main__), 70

main() (in module foremast.slacknotify.__main__), 72

make_elb_json() (foremast.elb.create_elb.SpinnakerELB method), 59

N

normalize_pipeline_name() (in module foremast.utils.pipelines), 82

normalize_url() (foremast.utils.gate.Gate method), 80

notify_slack_channel() (foremast.slacknotify.slack_notification.SlackNotification method), 72

P

post_message() (foremast.slacknotify.slack_notification.SlackNotification method), 72

post_pipeline() (foremast.pipeline.create_pipeline.SpinnakerPipeline method), 66

post_pipeline() (foremast.pipeline.create_pipeline.onetime.SpinnakerPipeline method), 68

post_slack_message() (in module foremast.utils.slack), 84

post_task() (in module foremast.utils.tasks), 84

prepare_app_pipeline() (in module foremast.runner), 91

prepare_infrastructure() (in module foremast.runner), 91

prepare_onetime_pipeline() (in module foremast.runner), 91

prepare_policy_template() (foremast.autoscaling_policy.create_policy.AutoScalingPolicy method), 46

PrimaryDNSRecordNotFound, 88

process_git_configs() (in module foremast.configs.prepare_configs), 55

process_runway_configs() (in module foremast.configs.prepare_configs), 55

promote_s3app() (foremast.runner.ForemastRunner method), 90

promote_s3app() (in module foremast.runner), 91

R

rebuild_pipelines() (in module foremast.runner), 91

remote_file() (foremast.utils.lookups.FileLookup method), 81

remove_all_lambda_permissions() (in module foremast.utils.awslambda), 73

render_wrapper() (foremast.pipeline.create_pipeline.SpinnakerPipeline method), 66

render_wrapper() (foremast.pipeline.create_pipeline_lambda.SpinnakerPipelineLambda method), 67

renumerate_stages() (in module foremast.pipeline.renumerate_stages), 68

RequiredKeyNotFound, 88

resource_action() (in module foremast.iam.resource_action), 63

S

S3ArtifactNotFound, 88

S3SharedBucketNotFound, 88

settings (foremast.autoscaling_policy.create_policy.AutoScalingPolicy attribute), 46

setup_lambda_api() (foremast.awslambda.api_gateway_event.api_gateway_event.Event.APIGateway method), 48

slack_notify() (foremast.runner.ForemastRunner method), 90

SlackNotification (class in foremast.slacknotify.slack_notification), 72

SNSSubscriptionDoesNotExist, 88

SNSTopicNotFound, 88

SpinnakerApp (class in foremast.app.create_app), 44

SpinnakerApplicationListError, 88

SpinnakerAppNotFound, 88

SpinnakerDns (class in foremast.dns.create_dns), 56

SpinnakerDnsCreationFailed, 88

SpinnakerELB (class in foremast.elb.create_elb), 58

SpinnakerElbNotFound, 89

SpinnakerError, 89

SpinnakerPipeline (class in foremast.pipeline.create_pipeline), 66

SpinnakerPipelineCreationFailed, 89

SpinnakerPipelineDeletionFailed, 89

SpinnakerPipelineLambda (class in foremast.pipeline.create_pipeline_lambda), 66

SpinnakerPipelineOnetime (class in foremast.pipeline.create_pipeline_onetime), 67

SpinnakerSecurityGroup (class in foremast.securitygroup.create_securitygroup), 71

SpinnakerSecurityGroupCreationFailed, 89

SpinnakerSecurityGroupError, 89

SpinnakerSubnetError, 89

SpinnakerTaskError, 89

SpinnakerTimeout, 89

SpinnakerVPCIDNotFound, 89

SpinnakerVPCNotFound, 89

splay_health() (in module foremast.elb.splay_health), 60

U

update_alias() (foremast.awslambda.awslambda.LambdaFunction method), 53

update_lambda_mappings() (foremast.awslambda.api_gateway_event.api_gateway_event.APIGateway method), 48

update_dns_zone_record() (in module foremast.utils.dns), 76

update_failover_dns_record() (in module foremast.utils.dns), 76

update_function_configuration() (foremast.awslambda.awslambda.LambdaFunction method), 53

V

validate_key_values() (in module foremast.consts), 87

W

wait_for_task() (in module foremast.utils.tasks), 84

warning_format() (in module foremast.utils.warn_user), 85

write_configs() (foremast.runner.ForemastRunner method), 90

write_variables() (in module foremast.configs.outputs), 54