Using Heat

OpenStack Orchestration on the Rapid Access Cloud

Introduction

Heat, the OpenStack Orchestration Service, is an OpenStack native service giving you the ability to describe your entire cloud environment in code, outlining all the resources and parameters for your cloud, then execute that code to automate provisioning multiple instances, volumes and other OpenStack resources. The following is all that is needed to provision a single `m1.small` instance with Floating IP:

```
heat_template_version: 2015-04-30
description: Simple template to deploy a single compute instance

resources:
 my_instance:
   type: OS::Nova::Server
   properties:
     key_name: my_key
     user_data_format: RAW
     image: Ubuntu 14.04
     flavor: m1.small
 my_instance_floating_ip:
   type: OS::Nova::FloatingIP
   properties:
     pool: nova
 my_instance_floating_ip_association:
   type: OS::Nova::FloatingIPAssociation
   properties:
     floating_ip: {get_resource: my_instance_floating_ip}
     server_id: {get_resource: my_instance}
```

Why Heat?

Heat will allow you to provision large complex cloud environments without needing to manually create each individual part, or by creating scripts that need to be updated or forked in order to automate the process across different environments. Once you have a configuration file, you can roll out the environment for development, testing, production by changing the input values. For example, a ‘simple’ web application that is load-balanced and/or built for high-availability may require the following components:

- 2x load-balancer/HA instances,
- 2x web-server instances,
- 2x application servers,
- 2x database servers,
- 1x volume to house the data,
- attach the volume to the database server,
- grab a floating IP address from your pool,
- assign it to the front-end load-balancer/HA cluster

Each of these separate items will need specific configuration and each instance will need to be identical to its pair. Since each item needs to be configured, a template allows you to list all the attributes and then treat your infrastructure as code, execute the code and launch all the individual elements.

Heat terminology

resource: individual components across OpenStack services, defined by type (e.g. instances, volumes, Floating IP addresses and the association with an instance, security groups, security group rules, etc.)

resource type: defines what the resource is and what properties it has. All resource types are listed here. Each type lists the associated attributes.

⚠ User should be logged into the Rapid Access Cloud to follow link above.

properties: the configuration settings for a given resource.

stack: a logical collection of resources that describes a single environment.

HOT: Heat Orchestration Template written in yaml.

Links

- OpenStack Heat Orchestration Template (HOT) documentation - http://docs.openstack.org/developer/heat/template_guide/hot_guide.html
- OpenStack Heat Python Client documentation - http://docs.openstack.org/developer/python-heatclient/man/heat.html
- NECTAR Heat documents - https://support.ehelp.edu.au/support/solutions/articles/6000055383-introduction-to-heat
- YAML Ain't Markup Language - http://yaml.org/

Provisioning your stack...

...with heat-pythonclient

1. Make sure you have the heat client installed:

```
$ sudo apt-get install heat-pythonclient
```

2. Once you have a template in a text file (simple.yaml) you can kick off your instance with:

```
$ heat stack-create -f simple.yaml myStack
```

(If you need more help using OpenStack via the command-line, see Command-line Tools in the Advanced Guide)

...with the dashboard

You can also use the Rapid Access Cloud dashboard (https://cloud.cybera.ca) to orchestrate your environment.

1. From the navigation menu on the left, select Orchestration -> Stacks:

blocked URL

From this screen you can Launch a stack, examine existing stacks, and delete stacks.

2. Choose Launch Stack:

blocked URL

3. Choose a template file (we have provided two at the end of this document: simple.yaml and shortstack.yaml - edit as you like). Note that Environment Source is not required. Press Next.

blocked URL

4. Name your stack and enter your password and click Launch.

⚠ Make sure your resource allocation has capacity for each of the items being launched (e.g. enough RAM, vCPUs, Floating IP addresses, etc.)

5. Once the stack has begun to launch, you can click on its name to see its topology, an overview of the stack, as well as any error messages encountered during the launch.

Caveats

You will find many example templates on the Internet, and it is certainly recommended that you make use of them to get an idea of how you can automate your environment. There are some things you need to remember however:

- Cybera's Rapid Access Cloud employs nova-network for networking services, so you must use `OS::Nova::FloatingIP and OS::Nova::FloatingIPAssociation` resource types to provision an IP address then associate it with your instance (see example at the beginning of the document). Any resources of the type `OS:Neutron::` are not supported in the Rapid Access Cloud.
- For compatibility reasons, you must include the property `user_data_format: RAW` for each resource of type `OS::Nova::Server`.