Contents

- 1 Background
- 2 Prepare the Images
- 3 Configure the OpenStack DHCP Domain
- 4 Heat Deployment
- 5 OpenShift Deployment
- 6 OpenShift Deployment Verification
- 7 Deploy Your First Application
- 8 Adding Nodes
- 9 Support
- 10 Credits
- 11 Authors

Background

This document provides users step-by-step instructions for deploying OpenShift Origin (OSO) v3.0 using OpenStack Heat. Heat is the main project in the OpenStack Orchestration program. It implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code. The current OSO Heat template supports deploying one Broker instance and one Node instance. The base OpenStack deployment was accomplished using Cisco's OpenStack Installer (COI) Havana release. However, the Heat deployment should work with any OpenStack distribution that uses Icehouse-1 and Neutron VLAN Provider Networking. Therefore, if you are using COI to manage your OpenStack deployment, you must use either the full_ha or compressed_ha scenario. This is because both scenarios use Neutron VLAN Provider Networking. Please file a feature request if you require support for a different Neutron networking model.

Prepare the Images

Images must be created for the Broker and Node instance. The images will be used to deploy running Broker and Node instances using the OpenStack Image Service (Glance), Compute Service (Nova) and Orchestration Service (Heat).

Log into a host that contains the following:

- OpenStack client packages (i.e. python-novaclient)
- Network connectivity to OpenStack API endpoints
- OpenStack credential file. (i.e. openrc). Here is a reference to the contents of an authentication file.

Install git and download Heat templates:

```
apt-get install -y git
# Note: The upstream openshift heat-template repo is still waiting for: https://review.openstack.c
# Use git clone https://github.com/openstack/heat-templates.git when the patch above is merged.
git clone -b oso_puppet_refactor https://github.com/danehans/heat-templates.git
```

Perform the following steps in the parent directory of heat-templates to build the images using diskimage-builder:

Download the diskimage-builder software and dependencies:

```
git clone https://github.com/openstack/diskimage-builder.git
apt-get install -y qemu-utils kpartx policycoreutils
```

Create environmental variables for the Broker image:

```
mkdir $HOME/tmp
export DIB_RELEASE=19
export ELEMENTS_PATH=heat-templates/openshift-origin/F19/elements
export TMP_DIR=$HOME/tmp
export DIB_IMAGE_SIZE=5
```

Create the Broker image. Be patient as it takes several minutes to build the image:

```
diskimage-builder/bin/disk-image-create --no-tmpfs -a amd64 vm fedora openshift-origin-broker -o F
```

If you have not done so already, load your credential file.

```
source /root/openrc
```

Note: A credential file can be avoided by using the necessary Glance flags to specify the auth URL, username, password, etc..

Add the newly created Broker image to Glance:

```
glance image-create --name F19-x86_64-openshift-origin-broker --is-public true --disk-format qcow2
```

Change the DIB_IMAGE_SIZE environmental variable for the Node image:

```
export DIB_IMAGE_SIZE=20
```

Create the Node image. Be patient as it may take several minutes to build the image:

```
diskimage-builder/bin/disk-image-create --no-tmpfs -a amd64 vm fedora openshift-origin-node -o F19
```

Add the newly created Node image to Glance:

```
glance image-create --name F19-x86_64-openshift-origin-node --is-public true --disk-format qcow2 -
```

Verify the Broker and Node images have been installed on Glance and has an active status:

```
glance image-list
+--------------------------------------+-----------------------------------------+--------------+-
| ID                                   | Name                                    | Disk Format |
+--------------------------------------+-----------------------------------------+--------------+-
| 102d07df-53e9-4499-b04d-32d9e217e2d1 | F19-x86_64-openshift-origin-broker      | qcow2       |
| 5a93c647-98c7-4b46-8461-ac17d3a1aee7 | F19-x86_64-openshift-origin-node        | qcow2       |
+--------------------------------------+-----------------------------------------+--------------+-
```

Follow the instructions in the SSH Key Injection Section of the Cisco Havana HA Manual Deployment Guide to create a Nova key-pair.

If Neutron networks have yet to be created, create your first tenant network now. Keep in mind this example uses Neutron Provider VLAN Networks which is the only supported networking model for the OSO Heat template. In our example, we use the admin tenant. Create additional networks as needed. **Note:** The *--tenant_id* flag is not specified in the following commands because we previously sourced our credential file.

```
neutron net-create public223 --provider:network_type vlan --provider:physical_network physnet1 --p
```

Create your first tenant subnet and associate it to the network you created in the previous step. The example below uses .10-.250 for Instance IP addresses. Modify the allocation-pool and dns_nameservers based on your deployment needs. Create additional networks as needed.

```
neutron subnet-create --name 223-subnet --allocation-pool start=192.168.223.10,end=192.168.223.250
```

Configure the OpenStack DHCP Domain

Nova uses metadata to manage the hostname of instances. By default the hostname of instances will be the name of the instance in the Nova boot command, followed by a period and the domain. The domain is either novalocal or openstacklocal by default, depending on the method used for accessing metadata. This domain **MUST** match the domain used within your OpenShift deployment. The example below sets the domain to example.com in /etc/nova/nova.conf:

```
vi /etc/nova/nova.conf
dhcp_domain=example.com
```

Restart the Nova API service:

```
service nova-api restart
```

The example below sets the domain to example.com in /etc/neutron/dhcp_agent.ini

```
vi /etc/neutron/dhcp_agent.ini
dhcp_domain=example.com
```

Restart the Neutron DHCP Agent:

```
service neutron-dhcp-agent restart
```

Heat Deployment

Since the Icehouse-1 release of Heat is required for the OSO template, Heat must be installed from source. At the time of this writing, the latest version of COI is Havana. If you have an existing Heat installation prior to 2014.1.b2, uninstall all Heat packages:

```
apt-get remove -y heat-common heat-api heat-api-cfn heat-api-cloudwatch heat-engine
```

If you are planning to deploy OpenStack using COI or you have just uninstalled the Heat packages from an existing COI deployment, remove Heat from the list of services that are managed by COI:

Comment-out heat_all from /etc/puppet/data/class_groups/controller.yaml

```
#heat_all
```

Install Heat from source:

```
apt-get install -y git
git clone https://github.com/openstack/heat.git
cd heat
git checkout 2014.1.b1
```

Install Heat source dependencies:

```
apt-get install python-dev python-pip libxml2-dev libxslt1-dev libsasl2-dev libsqlite3-dev libssl-
```

Install Heat:

```
pip install -r ~/heat/test-requirements.txt
python ~/heat/setup.py install
```

Create the Heat directories if they do not exist:

```
mkdir /etc/heat
mkdir /var/log/heat
chown heat:heat /etc/heat
chown heat:heat /var/log/heat
```

Copy the Heat config files:

```
cp -R ~/heat/etc/heat/* /etc/heat/.
```

Edit the /etc/heat/heat.conf file to include the following. Replace <CONTROLLER_IP> with the IP address of the Controller. Replace <CONTROLLER_VIP> with the VIP address of the Controller Cluster. **Note:** The example configuration below is based on the COI full_ha scenario. If you are not using the full_ha or compressed_ha scenario, use <CONTROLLER_IP> instead of <CONTROLLER_VIP> and follow the inline notes:

```
[DEFAULT]
heat_metadata_server_url=http://<CONTROLLER_VIP>:8000
heat_waitcondition_server_url=http://<CONTROLLER_VIP>:8000/v1/waitcondition
heat_watch_server_url=http://<CONTROLLER_VIP>8003
heat_stack_user_role=heat_stack_user
log_dir=/var/log/heat
# Comment-out rabbit_hosts if not using COI full_ha scenario
rabbit_hosts=control-server01:5672,control-server02:5672,control-server03:5672
rabbit_userid=openstack_rabbit_user
# Comment-out rabbit_ha_queues if not using COI full_ha scenario
rabbit_ha_queues=True
rabbit_password=openstack_rabbit_password

[database]
connection = mysql://heat:heat@<CONTROLLER_VIP>/heat
# Comment-out idle_timeout if not using COI full_ha scenario
idle_timeout=30

[ec2authtoken]
http://<CONTROLLER_VIP>:5000/v2.0/ec2tokens

[heat_api]
bind_host=<CONTROLLER_IP>

[heat_api_cfn]
bind_host=<CONTROLLER_IP>

[heat_api_cloudwatch]
bind_host=<CONTROLLER_IP>

[keystone_authtoken]
auth_host=<CONTROLLER_VIP>
auth_protocol=http
auth_uri=http://<CONTROLLER_VIP>:5000/v2.0/
```

```
admin_token=keystone_admin_token
admin_user=heat
admin_password=heat_pass
admin_tenant_name=services
```

Create the Heat database and necessary privileges:

```
mysql -u root -p
mysql> CREATE DATABASE heat;
mysql> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'localhost' DENTIFIED BY 'heat';
mysql> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'%' IDENTIFIED BY 'heat';
```

Start the Heat services:

```
/usr/bin/python /usr/local/bin/heat-api --config-file=/etc/heat/heat.conf &
/usr/bin/python /usr/local/bin/heat-api-cfn --config-file=/etc/heat/heat.conf &
/usr/bin/python /usr/local/bin/heat-api-cloudwatch --config-file=/etc/heat/heat.conf &
/usr/bin/python /usr/local/bin/heat-engine --config-file=/etc/heat/heat.conf &
```

Verify the Heat services are running:

```
 ps -ef | grep heat
```

OpenShift Deployment

At this point, you should have a functioning OpenStack deployment that includes Heat (2014.1.b2 or later) and that your OpenShift Broker and Node images have been successfully uploaded to Glance. Now simply issue the *heat stack-create* command to deploy your OpenShift Origin environment:

```
heat stack-create <STACK_NAME --template-file=<HEAT_TEMPLATE_FILE_PATH> \
--parameters="KeyName=<KEY_NAME>;Prefix=<DOMAIN_NAME>;UpstreamDNS=<UPSTREAM_DNS>;\
UpstreamNTP=<UPSTREAM_NTP>;NetID=<NEUTRON_NET_ID>;\
SubnetID=<NEUTRON_SUBNET_ID>;BrokerHostname=<BROKER_NAME>;NodeHostname=<NODE_NAME>;Username=<USERN
PuppetModuleURL=https://github.com/danehans/puppet-openshift_origin.git;PuppetModuleBranch=systemd
```

Here is an overview of the parameters contained within the *heat stack-create* command:

- <STACK_NAME>: A unique name for the OpenShift Heat stack.
- <HEAT_TEMPLATE_FILE_PATH>: The file path to the Heat OpenShift.template. If you are following these directions, the path should be: /root/heat-templates/openshift-origin/F19/OpenShift.template.
- <KEY_NAME>: The name of the Nova key-pair created earlier in this document. (i.e. admin-key)
- <DOMAIN_NAME>: The domain name that will be used by your OpenShift deployment. Defaults to example.com
- <UPSTREAM_DNS>: The IP address(es) of the upstream DNS serves used by your Broker instance for name resolution (outside of <DOMAIN_NAME>). Defaults to Google DNS (8.8.8.8).
- <UPSTREAM_NTP>: IP address of the upstream NTP server. Defaults to 'clock.redhat.com iburst'. Note: iburst must be included in your NTP server entry and therefore a ' ' is required for this parameter.
- <NEUTRON_NET_ID>: The ID of the Neutron network that will be used to spawn Broker and Node instances on. You can obtain the Neutron Net ID from the *neutron net-list* command.
- <NEUTRON_SUBNET_ID>: The ID of the Neutron subnet that will be used to spawn Broker and Node instances on. You can obtain the Neutron Subnet ID from the *neutron subnet-list* command.
- <BROKER_NAME>: The name that will be appended to <DOMAIN_NAME> and used as the hostname of the Broker instance. Defaults to openshift.brokerinstance.novalocal.

- <NODE_NAME>: The name that will be appended to <DOMAIN_NAME> and used as the hostname of the Node instance. Defaults to openshift.nodeinstance.novalocal.
- <USERNAME>: The username of the admin account for managing the OpenShift environment. Defaults to openshift.
- <PASSWORD>: The password of the admin account for managing the OpenShift environment. Defaults to password.
- Remove PuppetModuleURL=https://github.com/danehans/puppet-openshift_origin.git;PuppetModuleBranch=systemd_vim from the command above when upstream pull requests are merged.

Here is an example of the *heat stack-create* command with the required parameters supplied:

```
heat stack-create oso-stack --template-file=/root/heat-templates/openshift-origin/F19/OpenShift.te
--parameters="KeyName=admin-key;Prefix=example.com;UpstreamDNS=192.168.26.186;\
UpstreamNTP='ntp.esl.cisco.com iburst';NetID=df4a1975-aa4f-40f5-bd86-f1a10ccbde46;\
SubnetID=ce313fbb-3379-4b47-8949-54c6f54c962a;BrokerHostname=broker02;NodeHostname=node02"
```

OpenShift Deployment Verification

The OpenShift Heat stack takes anywhere from 20-30 minutes to be deployed and fully configured, so please be patient. You can observe the status of the deployment with the following steps:

Verify the progress of the OpenShift stack:

```
heat stack-list
+--------------------------------------+------------+--------------------+----------------------+
| id                                   | stack_name | stack_status       | creation_time        |
+--------------------------------------+------------+--------------------+----------------------+
| 7c366d13-08ec-41b1-aa3d-6f9e5a197140 | oso-stack  | CREATE_IN_PROGRESS | 2014-02-12T22:37:32Z |
+--------------------------------------+------------+--------------------+----------------------+
```

Review the details of the OpenShift stack:

```
 heat stack-show <STACK_NAME>
+----------------------+--------------------------------------------------------------------------------
| Property             | Value
+----------------------+--------------------------------------------------------------------------------
| capabilities         | []
| creation_time        | 2014-03-07T18:41:07Z
| description          | Template for setting up an OpenShift Origin environment
| disable_rollback     | True
| id                   | 349e7128-cd15-4333-91ed-dff4b9589a46
| links                | http://192.168.220.41:8004/v1/92797818f9724adc8e7b7695028f2a4d/stacks/osc
| notification_topics  | []
| outputs              | [
|                      |   {
|                      |     "output_value": "nameserver 192.168.223.11",
|                      |     "description": "Entry to insert into /etc/resolv.conf for application
|                      |     "output_key": "NameServerEntry"
|                      |   },
|                      |   {
|                      |     "output_value": "https://192.168.223.11/console",
|                      |     "description": "URL for OpenShift Origin cConsole",
|                      |     "output_key": "OpenShiftConsole"
|                      |   }
|                      | ]
| parameters           | {
|                      |   "Username": "openshift",
|                      |   "BrokerFlavor": "m1.small",
```

```
|                      |   "NodeHostname": "node",
|                      |   "NetID": "df4a1975-aa4f-40f5-bd86-f1a10ccbde46",
|                      |   "UpstreamNTP": "'ntp.esl.cisco.com iburst'",
|                      |   "DevMode": "false",
|                      |   "PuppetModuleBranch": "master",
|                      |   "AWS::StackName": "oso-stack",
|                      |   "AWS::StackId": "arn:openstack:heat::92797818f9724adc8e7b7695028f2a4d:s
|                      |   "KeyName": "admin-key",
|                      |   "PuppetModuleURL": "https://github.com/openshift/puppet-openshift_origi
|                      |   "UpstreamDNS": "192.168.26.186",
|                      |   "SubnetID": "ce313fbb-3379-4b47-8949-54c6f54c962a",
|                      |   "NodeFlavor": "m1.small",
|                      |   "Password": "password",
|                      |   "AWS::Region": "ap-southeast-1",
|                      |   "Prefix": "example.com",
|                      |   "BrokerHostname": "broker"
|                      | }
| stack_name           | oso-stack
| stack_status         | CREATE_COMPLETE
| stack_status_reason  | Stack create completed successfully
| template_description | Template for setting up an OpenShift Origin environment
| timeout_mins         | 60
| updated_time         | 2014-03-07T19:04:06Z
+----------------------+-------------------------------------------------------------------------
```

Verify the Neutron security-group has been created for the OpenShift stack:

```
neutron security-group-list
+--------------------------------------+-----------------------------------------------+---------
| id                                   | name                                          | descript
+--------------------------------------+-----------------------------------------------+---------
| 7a342261-a35d-43d6-8b74-c73ae9741e79 | oso-stack-OpenShiftOriginSecurityGroup-2c6rzmfgug65  | Op
| a8082e6e-8203-48bc-a211-31b38dd8620b | default                                       | default
+--------------------------------------+-----------------------------------------------+---------
```

Verify the Neutron Broker/Node ports has have created for the OpenShift stack. '*Note:* You will see an additional port that has been provisioned by Neutron for the DHCP Agent.

```
neutron port-list
+--------------------------------------+-----------------------------+-------------------+-------
| id                                   | name                        | mac_address       | fixed_
+--------------------------------------+-----------------------------+-------------------+-------
| 45d81f77-fad9-48e7-97d6-1dbe3b7d9102 | oso-stack-NodePort-lmqxyringm6p  | fa:16:3e:c7:ca:93 | {"
| 8665eab6-d267-463d-9723-46c2d2d8bf54 | oso-stack-BrokerPort-3zy5ng4hcr73| fa:16:3e:b2:bb:36 | {"
| abb46c5d-08cf-4d17-917c-57b6cba937d7 |                           | fa:16:3e:22:f6:db    |  {"subn
+--------------------------------------+-----------------------------+-------------------+-------
```

Verify the Broker instance has been spawned and shows an **ACTIVE** status. **Note:** The Node instance does not get spawned until the Broker has been spawned and is fully configured:

```
nova list
+--------------------------------------+----------------------------------+--------+------------+-
| ID                                   | Name                             | Status | Task State |
+--------------------------------------+----------------------------------+--------+------------+-
| 439cca5f-9b61-46c8-90d4-065d315af15e | oso-stack-BrokerInstance-eml774uhi2hg  | ACTIVE | None   |
+--------------------------------------+----------------------------------+--------+------------+-
```

You should then be able to SSH into your Broker instance. Keep in mind that you need to specify the SSH private key that was used to create the Nova key-pair. ec2-user is the default user name. You can change the default user name in heat.conf (instance_user=ec2-user).

```
ssh -i ~/.ssh/id_rsa ec2-user@<BROKER_IP>
sudo -i
```

You can view the installation log at /var/log/configure_openshift.log

```
tail -f /var/log/configure_openshift.log
```

Note:' You can safely ignore the following warning messages that may appear during your Puppet run:
Warning: Config file /etc/puppet/hiera.yaml not found, using Hiera defaults
Warning: Augeas[network-scripts](provider=augeas): Loading failed for one or more files, see debug for /augeas//error output

The puppet run should complete with the following message:

```
Notice: Finished catalog run in xxx seconds
```

You can also view the Heat provisioning log.

```
less /var/log/heat-provision.log
```

Note: The last line in the log should be *Provision done: 2014-03-07 18:47:09.791809*

The broker is now successfully deployed by Heat. You can repeat these verification steps for the Node instance if you would like. Otherwise, wait 10-15 minutes for Heat to complete the rest of the stack deployment and issue the *heat stack-list* or *heat stack-show <STACK_NAME>* commands to verify the successful completion of the stack:

```
heat stack-list
+--------------------------------------+------------+--------------------+----------------------+
| id                                   | stack_name | stack_status       | creation_time        |
+--------------------------------------+------------+--------------------+----------------------+
| 349e7128-cd15-4333-91ed-dff4b9589a46 | oso-stack  | CREATE_COMPLETE    | 2014-03-07T18:41:07Z |
+--------------------------------------+------------+--------------------+----------------------+
```

Deploy Your First Application

At this point, your OpenShift stack should have been successfully deployed:

```
heat stack-list
+--------------------------------------+------------+--------------------+----------------------+
| id                                   | stack_name | stack_status       | creation_time        |
+--------------------------------------+------------+--------------------+----------------------+
| 349e7128-cd15-4333-91ed-dff4b9589a46 | oso-stack  | CREATE_COMPLETE    | 2014-03-07T18:41:07Z |
+--------------------------------------+------------+--------------------+----------------------+
```

If not, do not proceed with this section and follow the Deployment Verification for troubleshooting assistance. Otherwise, follow the Deploy Your First Application Section of the OpenShift Automated Deployment Guide.

Adding Nodes

Follow the Node Deployment Section of the OpenShift Automated Deployment Guide for adding nodes to your deployment.

Support

OpenShift Mailer

Credits

This document is based on the following:

- OpenStack Heat Wiki [1]
- OpenStack Documentation [[2]
- OpenShift Origin Comprehensive Deployment Guide [3]
- OpenShift Origin User?s Guide [4]
- OpenShift Example PHP Readme [5]

Authors

Daneyon Hansen