

## Contents

- [1 Background](#)
- [2 Dependencies](#)
- [3 OpenStack Cloud Preparation](#)
- [4 Broker Deployment](#)
- [5 Node Deployment](#)
- [6 Deploy Your First Application](#)
- [7 Support](#)
- [8 Credits](#)
- [9 Authors](#)

## Background

This document provides users step-by-step instructions for automating an OpenShift Origin v3.0 deployment. Deployment automation is accomplished through an [installation script](#), which makes use of [Puppet](#). The reference deployment is based on an OpenStack cloud infrastructure and includes specific steps for preparing the OpenStack environment for OpenShift. However, the automated deployment can work in non-OpenStack environments. Although most of the deployment is automated, a few manual steps must be taken to prepare and test the OpenShift deployment.

## Dependencies

The following dependencies must be met for a successful OpenShift Origin automated deployment:

**OpenShift Origin-** Version 3.0.1 was used for this guide.

**Internet Access-** Used for pulling packages from repositories.

**Operating System-** Fedora 19 with sudo privileges. **Note:** This requirement is satisfied in the [OpenStack Cloud Preparation Section](#) by installing a Fedora 19 image on the OpenStack Image Service (i.e. Glance).

**OpenStack-** The OpenStack environment used for this guide was based on the [Cisco Havana HA Manual Deployment Guide](#). The [Cisco Havana OpenStack Installer](#) should also work, but has not been verified at this time.

## OpenStack Cloud Preparation

As previously mentioned, the reference deployment uses an OpenStack cloud environment to host OpenShift Origin. The following steps must be taken to prepare the OpenStack environment for OpenShift:

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.

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..

## OpenShift\_Origin\_Automated\_Deployment\_Guide

Install the Fedora 19 image into Glance:

```
glance image-create --name "fedora_19_x86_64" --disk-format qcow2 --container-format bare --is-public --copy-from http://download.fedoraproject.org/pub/fedora/linux/releases/19/Images/x86_64/Fedora
```

Verify the `fedora_19_x86_64` image has been installed on Glance and has an active status:

```
glance image-list
```

ID	Name	Disk Format	Container Format
ee12ba5d-4a41-4459-8047-684d4ff07db1	fedora_19_x86_64	qcow2	bare

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

Configure Neutron to allow the necessary OpenShift traffic through the firewall. Keep in mind the OpenStack Havana HA reference deployment uses Provider Networking Extensions. When provider networking is used, all inbound traffic will be blocked. This includes traffic between the broker and node instances. Additional information about OpenStack Neutron Provider Networking Extensions can be found [here](#). The example Neutron security group rules below are quite permissive. It is recommended to use more restrictive rules for a production deployment:

```
neutron security-group-rule-create default --protocol udp --port-range-min 53 --port-range-max 53
neutron security-group-rule-create default --protocol tcp --port-range-min 53 --port-range-max 53
neutron security-group-rule-create default --protocol tcp --port-range-min 80 --port-range-max 80
neutron security-group-rule-create default --protocol tcp --port-range-min 443 --port-range-max 443
neutron security-group-rule-create default --protocol tcp --port-range-min 8443 --port-range-max 8443
neutron security-group-rule-create default --protocol tcp --port-range-min 8000 --port-range-max 8000
neutron security-group-rule-create default --protocol tcp --port-range-min 8161 --port-range-max 8161
neutron security-group-rule-create default --protocol tcp --port-range-min 8080 --port-range-max 8080
neutron security-group-rule-create default --protocol tcp --port-range-min 8181 --port-range-max 8181
```

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
```

## OpenShift\_Origin\_Automated\_Deployment\_Guide

Boot the broker instance. **Note:** The <INSTANCE\_NAME> should be the hostname of your Broker instance (i.e. broker).

```
nova boot --image fedora_19_x86_64 --flavor m1.small --key_name <NOVA_KEYPAIR_NAME> --nic net-id=<
```

- Replace <NOVA\_KEYPAIR\_NAME> with the name of your Nova SSH keypair (i.e. test-key)
- Replace <NEUTRON\_TENANT\_NETWORK\_ID> with the ID of the Neutron network that is provided to you as a tenant. Use the *neutron net-list* command to obtain your neutron network ID.
- Replace <INSTANCE\_NAME> with the name of your broker instance.

Verify that the status of the OpenShift Broker instance is **ACTIVE** and record the IP address of the instance:

```
nova show <INSTANCE_NAME>
```

Property	Value
status	ACTIVE
updated	2014-01-29T21:23:17Z
OS-EXT-STS:task_state	None
OS-EXT-SRV-ATTR:host	compute-server02
key_name	test-key
image	fedora_19_x86_64 (ee12ba5d-4a41-4459-8047-684d4ff07db1)
hostId	61d3bd708b013e0ebdfc5e910d22bc8d714e9e84a6571106656242d6
OS-EXT-STS:vm_state	active
OS-EXT-SRV-ATTR:instance_name	instance-0000011e
OS-SRV-USG:launched_at	2014-01-29T21:08:31.000000
OS-EXT-SRV-ATTR:hypervisor_hostname	compute-server02.dmz-pod2.lab
flavor	m1.small (2)
id	85942f51-df9c-458f-9ad2-493a7388b868
security_groups	[[{'u'name': u'default'}]]
OS-SRV-USG:terminated_at	None
public223 network	192.168.223.10
user_id	1d4b7a90b55d421cbba57c80d8fbb97
name	broker
created	2014-01-29T21:08:23Z
tenant_id	7a0e47f62ee047f789d04408c07e9f32
OS-DCF:diskConfig	MANUAL
metadata	{}
os-extended-volumes:volumes_attached	[]
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	1
OS-EXT-AZ:availability_zone	nova
config_drive	

Follow the same steps as above to boot your first OpenShift Node instance. Make sure to change the <INSTANCE\_NAME> to the hostname of the Node instance (i.e. node).

## Broker Deployment

After the Broker instance has been successfully spawned on your OpenStack cloud. Log into the instance and enter sudo mode:

```
ssh -i <SSH_PRIVATE_KEY> fedora@<BROKER_INSTANCE_IP>
sudo -i
```

## OpenShift\_Origin\_Automated\_Deployment\_Guide

- Replace <SSH\_PRIVATE\_KEY> with the SSH private key used to create your Nova keypair. Refer to the [SSH Key Injection Section](#) of the Cisco Havana HA Manual Deployment Guide for additional details on this step.
- Replace <BROKER\_INSTANCE\_IP> with the IP address of the Broker instance. You can issue the `nova list` or `nova show <INSTANCE_NAME>` to obtain the IP address of the Broker instance.

Set the installation type.

```
export INSTALL_TYPE=broker
```

Set the Domain Name prefix that your OpenShift Origin deployment will use. Replace `example.com` with your Domain Name prefix.

```
export PREFIX=example.com
```

Set the upstream DNS server that your OpenShift Origin deployment will use. **Note:** Google DNS (8.8.8.8) is used by default.

```
export UPSTREAM_DNS=8.8.8.8
```

Set the upstream NTP server that your OpenShift Origin deployment will use. **Note:** RedHat NTP (`clock.redhat.com`) is used by default.

```
export UPSTREAM_NTP=ntp.corp.com
```

Set the OpenShift Origin authentication credentials. **Note:** `openshift/password` are the default `OSO_USERNAME/OSO_PASSWORD`.

```
export OSO_USERNAME=admin
export OSO_PASSWORD=ChangeMe
```

Set the Ethernet interface that is used by your Broker server for networking connectivity. **Note:** `eth0` is used by default.

```
export ETH_DEV=eth0
```

Download and install the installation script:

```
bash <(curl -fsS https://raw.githubusercontent.com/danehans/puppet-openshift_origin/install_scripts/install_
```

**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 view the installation log at `/var/log/configure_openshift.log`

## Node Deployment

To successfully deploy a Node server, you **MUST** obtain two pieces of data from the Broker server. Log into the Broker instance and enter sudo mode:

```
ssh -i <SSH_PRIVATE_KEY> fedora@<BROKER_INSTANCE_IP>
sudo -i
```

- Replace <SSH\_PRIVATE\_KEY> with the SSH private key used to create your Nova keypair. Refer to the [SSH Key Injection Section](#) of the Cisco Havana HA Manual Deployment Guide for additional details on this step.
- Replace <BROKER\_INSTANCE\_IP> with the IP address of the Broker instance. You can issue the `nova l`

Obtain the DNSSEC security key. Replace example.com with the Domain Name used for your OpenShift deployment:

```
export PREFIX=example.com
cat /var/named/K${PREFIX}.*.key | awk '{print $8}'
```

Print the network interface settings. Replace <ETH\_DEV> with the name of the ethernet device used for network connectivity (i.e. eth0). Copy the IP address, as you will need it for your Node server deployment.

```
ifconfig <ETH_DEV>
```

You can now log out of your broker server.

After the Node instance has been successfully spawned on your OpenStack cloud. Log into the instance and enter sudo mode:

```
ssh -i <SSH_PRIVATE_KEY> fedora@<NODE_INSTANCE_IP>
sudo -i
```

- Replace <SSH\_PRIVATE\_KEY> with the SSH private key used to create your Nova keypair. Refer to the [SSH Key Injection Section](#) of the Cisco Havana HA Manual Deployment Guide for additional details on this step.
- Replace <NODE\_INSTANCE\_IP> with the IP address of the Node instance. You can issue the `nova list` or `nova show <INSTANCE_NAME>` to obtain the IP address of the Node instance.

Set the DNSSEC security key that you copied from your Broker server. Replace <KEY\_FROM\_BROKER> with the actual key.

```
export DNS_SEC_KEY=<KEY_FROM_BROKER>
```

Set the installation type.

```
export INSTALL_TYPE=node
```

Set the Domain Name prefix that your OpenShift Origin deployment will use. Replace example.com with your Domain Name prefix.

```
export PREFIX=example.com
```

## OpenShift\_Origin\_Automated\_Deployment\_Guide

Set the upstream DNS server that your OpenShift Origin deployment will use. **Note:** Google DNS (8.8.8.8) is used by default.

```
export UPSTREAM_DNS=8.8.8.8
```

Set the upstream NTP server that your OpenShift Origin deployment will use. **Note:** RedHat NTP (clock.redhat.com) is used by default.

```
export UPSTREAM_NTP=ntp.corp.com
```

Set the Ethernet interface that is used by your Broker server for networking connectivity. **Note:** eth0 is used by default.

```
export ETH_DEV=eth0
```

Set the IP address of your Broker server. Replace <BROKER\_IP\_ADDRESS> with the actual IP address of the Broker server.

```
export BROKER_IP=<BROKER_IP_ADDRESS>
```

Download and install the installation script:

```
bash <(curl -fsS https://raw.githubusercontent.com/danehans/puppet-openshift_origin/install_scripts/install_
```

**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 view the installation log at /var/log/configure\_openshift.log

## Deploy Your First Application

After Puppet successfully completes the deployment of the Broker and Node instances, follow the proceeding steps to verify the operation of your OpenShift environment:

Open a browser and go to the OpenShift web console. The URL is the FQDN of your Broker instance:

```
https://broker.example.com/
```

**Note:** The host being used for this step must resolve the Broker's FQDN. Therefore, create a DNS A record for the Broker instance in your upstream DNS or create the mapping in your local host file. Additional details for configuring local name resolution can be found [here](#).

If prompted, make sure you trust the site's SSL certificate. You will next be prompted for a username and password. This is the `OSO_USERNAME` and `OSO_PASSWORD` from the environmental variable settings of your Broker instance.

### **Figure 1: OpenShift Web Console Homepage**



After successfully authenticating to the OpenShift Web Console, click on the Settings tab and create a Domain name in the Namespace section. This domain name will be appended to applications that are created (i.e. application\_name-openshift.example.com).

**Figure 2: OpenShift Web Console Configure Domain**



Optionally, you can upload your personal SSH public (.pub) key for secure password-less access to your OpenShift applications.

**Figure 3: OpenShift Web Console Configure SSH**



Next, use the RHC Client to create a test application. The RHC client was automatically installed on the Broker instance or you can install the RHC client on a separate host that has connectivity to the Broker API endpoint. Detailed instructions for installing RHC on a separate host can be found [here](#).

Run RHC setup. Replace <BROKER\_FQDN> with the FQDN of the Broker instance (i.e. broker.example.com). You will next be prompted for a username and password. This is the OSO\_USERNAME and OSO\_PASSWORD from the environmental variable settings of your Broker instance.

```
rhc setup --server <BROKER_FQDN>
```

Although it's not required, it's good practice to create an application directory used to contain the OpenShift applications you create:

```
cd ~
mkdir apps
cd apps
```

By default, application names created in OpenShift must resolve (i.e. cake-openshift.example.com). This is accomplished by default when using the RHC Client on the Broker instance. When using RHC from another host, a DNS A record must be created in your upstream DNS server or in /etc/hosts before you create an application. The IP address of the Node instance is used to resolve the application's FQDN. Additional details for configuring local name resolution can be found [here](#).

Create an application. In the example below, we will create a PHP application named cake that uses the php-5.5 cartridge. When the application has been successfully created, you should receive a message stating your application <APPLICATION\_NAME> is now available. Use the *rhc show-app* <APPLICATION\_NAME> for more details about your application.

```
rhc app create -a cake -t php-5.5
```

Add a database backend to your application. The example below adds a MariaDB database using the mariadb-5.5 cartridge to an application named cake (created in the previous step).

```
rhc cartridge add -a cake -c mariadb-5.5
```

Add the upstream Cake repo. **Note:** You will be automatically placed in a vi session during the git pull command. You can type <esc> : q to safely exit the vi session.

```
cd cake
git remote add upstream -m master git://github.com/openshift/cakephp-example.git
git pull -s recursive -X theirs upstream master
```



Push the code changes to your application running on the Node instance:

```
git push
```

Lastly, open your web browser and access the CakePHP application.

```
http://cake-openshift.example.com
```

**Figure 4: CakePHP Test Application Homepage**



## Support

[OpenShift Mailer](#)

## Credits

This document is based on the following:

- [OpenShift Origin Comprehensive Deployment Guide \[1\]](#)
- [OpenShift Origin User's Guide \[2\]](#)
- [OpenShift Example PHP Readme \[3\]](#)

## Authors

Daneyon Hansen