



OPNFV Installation instruction for the template-os-nosdn-nofeature-ha scenario

Release arno.2015.1.0 (2356f56)

OPNFV

August 12, 2016

1	Abstract	1
2	Introduction	3
3	Preparation	5
3.1	Preparing your jumphost to install by script	5
3.2	Preparing your jumphost using an ISO image	5
3.3	Additional preparations	6
4	Hardware requirements	7
4.1	The Pharos Lab	7
4.2	Virtual deployment hardware requirements	7
5	<scenario> software installation and deployment	9
5.1	Install jumphost	9
5.2	Boot the Node Servers	10
5.3	Install additional Plugins/Features on the FUEL node	10
5.4	Create an OpenStack Environment	10
6	Configure the network environment	13
7	Select Hypervisor type	15
8	Enable Plugins	17
9	Allocate nodes to environment and assign functional roles	19
10	OPTIONAL - Set Local Mirror Repos	21
11	Target specific configuration	23
12	Verify Networks	25
13	Deploy Your Environment	27
14	Installation health-check	29
15	References	31
15.1	OPNFV	31
15.2	OpenStack	31
15.3	OpenDaylight	31
15.4	Fuel	31
15.5	Fuel in OPNFV	31

ABSTRACT

This document describes how to install the <template> scenario of the Colorado release of OPNFV.

The <template> scenario will use Fuel as a deployment tool, installing and deploying an OPNFV based OpenStack in high availability on a Pharos compliant infrastructure.

INTRODUCTION

This document provides guidelines on how to install and configure the <template> scenario of OPNFV including required software and hardware configurations.

Although the available installation options give a high degree of freedom in how the system is set-up, including architecture, services and features, etc., said permutations may not provide an OPNFV compliant reference architecture. This instruction provides a step-by-step guide that results in an OPNFV Compliant deployment.

The audience of this document is assumed to have good knowledge in networking and Unix/Linux administration.

PREPARATION

Before starting the installation of the <scenario> scenario some preparation must be done. You may choose to install the <scenario> scenario using an ISO image, or executing the installation from a prepared jumphost.

3.1 Preparing your jumphost to install by script

To deploy the <scenario> scenario from a script you will need to prepare the jumphost with a compatible operating system. Prepare your jumphost running CentOS 7 with libvirt running on it. You may then install the RDO Release RPM:

```
sudo yum install -y https://www.rdoproject.org/repos/rdo-release.rpm  
opnfv-apex-{version}.rpm
```

The RDO Project release repository is needed to install OpenVSwitch, which is a dependency of opnfv-apex. If you do not have external connectivity to use this repository you need to download the OpenVSwitch RPM from the RDO Project repositories and install it with the opnfv-apex RPM.

3.2 Preparing your jumphost using an ISO image

An alternative to preparing your own jumphost id to use a <scenario> ISO image as a boot image. Download or build the ISO image according to the following instructions.

3.2.1 Retrieving the ISO image

If you choose to install the <scenario> scenario from an ISO image you must first retrieve the <template-containing>.iso image of the Colorado release. This can be found can be found at <hyperlink required>.

3.2.2 Building the ISO image

Alternatively, you may choose to build the Fuel .iso from source by cloning the opnfv/fuel git repository. To retrieve the repository for the Colorado release use the following command:

```
$ git clone https://gerrit.opnfv.org/gerrit/fuel
```

Check-out the Colorado release tag to set the HEAD to the baseline required to replicate the Colorado release:

```
$ git checkout colorado.1.0
```

Go to the fuel directory and build the .iso:

```
$ cd fuel/build; make all
```

For more information on how to build, please see *Reference: 14*

3.2.3 Booting from the ISO image

Mount the <scenario> ISO file/media as a boot device on the jump host server. If all your hardware preparation is complete at this time you should reboot the jumphost to establish the deployment server.

3.3 Additional preparations

Additionally, familiarize yourself with Fuel by reading the following documents:

- Fuel planning guide, please see *Reference: 8*
- Fuel user guide, please see *Reference: 9*
- Fuel operations guide, please see *Reference: 10*
- Fuel Plugin Developers Guide, please see *Reference: 11*

Prior to installation, a number of deployment specific parameters must be collected, those are:

1. Provider sub-net and gateway information
2. Provider VLAN information
3. Provider DNS addresses
4. Provider NTP addresses
5. Network overlay you plan to deploy (VLAN, VXLAN, FLAT)
6. How many nodes and what roles you want to deploy (Controllers, Storage, Computes)
7. Monitoring options you want to deploy (Ceilometer, Syslog, etc.).
8. Other options not covered in the document are available in the links above

This information will be needed for the configuration procedures provided in this document.

HARDWARE REQUIREMENTS

4.1 The Pharos Lab

Hardware requirements for OPNFV infrastructures are specified by the Pharos project. The Pharos project provides an OPNFV hardware specification for configuring your hardware at: <http://artifacts.opnfv.org/pharos/docs/pharos-spec.html> .

4.2 Virtual deployment hardware requirements

To perform a virtual deployment of an OPNFV scenario established different hardware requirements. The server requirements for this type of deployment are outlined in the <missing spec>.

<SCENARIO> SOFTWARE INSTALLATION AND DEPLOYMENT

This section describes the installation of the OPNFV installation server (jumphost) as well as the deployment of the <scenario> OPNFV reference platform stack across a server cluster.

5.1 Install jumphost

If you have not already done so, prepare your jumphost according the instructions in [_#ref_Preparation](#) this can be done using an ISO image with the following commands

1. Mount the <scenario> ISO file/media as a boot device to the jump host server.
2. Reboot the jump host to establish the jumphost server.
 - The system now boots from the ISO image.
 - Select “Fuel Install (Static IP)”
 - Press [Enter].
3. Wait until screen Fuel setup is shown (Note: This can take up to 30 minutes).

At this time you can continue with the installation according to the following instructions.

1. In the “Fuel User” section - Confirm/change the default password
 - Enter “admin” in the Fuel password input
 - Enter “admin” in the Confirm password input
 - Select “Check” and press [Enter]
2. In the “Network Setup” section - Configure DHCP/Static IP information for your FUEL node - For example, ETH0 is 10.20.0.2/24 for FUEL booting and ETH1 is DHCP in your corporate/lab network.
 - Configure eth1 or other network interfaces here as well (if you have them present on your FUEL server).
3. In the “PXE Setup” section - Change the following fields to appropriate values (example below):
 - DHCP Pool Start 10.20.0.3
 - DHCP Pool End 10.20.0.254
 - DHCP Pool Gateway 10.20.0.2 (IP address of Fuel node)
4. In the “DNS & Hostname” section - Change the following fields to appropriate values:
 - Hostname
 - Domain
 - Search Domain

- External DNS
 - Hostname to test DNS
 - Select <Check> and press [Enter]
5. OPTION TO ENABLE PROXY SUPPORT - In the “Bootstrap Image” section, edit the following fields to define a proxy. (**NOTE:** cannot be used in tandem with local repository support)
 - Navigate to “HTTP proxy” and enter your http proxy address
 - Select <Check> and press [Enter]
 6. In the “Time Sync” section - Change the following fields to appropriate values:
 - NTP Server 1 <Customer NTP server 1>
 - NTP Server 2 <Customer NTP server 2>
 - NTP Server 3 <Customer NTP server 3>
 7. Start the installation.
 - Select Quit Setup and press Save and Quit.
 - Installation starts, wait until the login screen is shown.

5.2 Boot the Node Servers

After the Fuel Master node has rebooted from the above steps and is at the login prompt, you should boot the Node Servers (Your Compute/Control/Storage blades (nested or real) with a PXE booting scheme so that the FUEL Master can pick them up for control.

1. Enable PXE booting
 - For every controller and compute server: enable PXE Booting as the first boot device in the BIOS boot order menu and hard disk as the second boot device in the same menu.
2. Reboot all the control and compute blades.
3. Wait for the availability of nodes showing up in the Fuel GUI.
 - Connect to the FUEL UI via the URL provided in the Console (default: <https://10.20.0.2:8443>)
 - Wait until all nodes are displayed in top right corner of the Fuel GUI: Total nodes and Unallocated nodes.

5.3 Install additional Plugins/Features on the FUEL node

1. SSH to your FUEL node (e.g. `root@10.20.0.2` pwd: `r00tme`)
2. Select wanted plugins/features from the `/opt/opnfv/` directory.
3. Install the wanted plugin with the command “`fuel plugins --install /opt/opnfv/<plugin-name>-<version>.<arch>.rpm`” Expected output: “Plugin was successfully installed.”

5.4 Create an OpenStack Environment

1. Connect to Fuel WEB UI with a browser (default: <https://10.20.0.2:8443>) (login admin/admin)

2. Create and name a new OpenStack environment, to be installed.
3. Select “<Liberty on Ubuntu 14.04>” and press <Next>
4. Select “compute virtulization method”.
 - Select “QEMU-KVM as hypervisor” and press <Next>
5. Select “network mode”.
 - Select “Neutron with ML2 plugin”
 - Select “Neutron with tunneling segmentation” (Required when using the ODL or ONOS plugins)
 - Press <Next>
6. Select “Storage Back-ends”.
 - Select “Ceph for block storage” and press <Next>
7. Select “additional services” you wish to install.
 - Check option “Install Ceilometer (OpenStack Telemetry)” and press <Next>
8. Create the new environment.
 - Click <Create> Button

CONFIGURE THE NETWORK ENVIRONMENT

1. Open the environment you previously created.
2. Open the networks tab and select the “default Node Networks group to” on the left pane.
3. Update the Public network configuration and change the following fields to appropriate values:
 - CIDR to <CIDR for Public IP Addresses>
 - IP Range Start to <Public IP Address start>
 - IP Range End to <Public IP Address end>
 - Gateway to <Gateway for Public IP Addresses>
 - Check <VLAN tagging>.
 - Set appropriate VLAN id.
4. Update the Storage Network Configuration
 - Set CIDR to appropriate value (default 192.168.1.0/24)
 - Set IP Range Start to appropriate value (default 192.168.1.1)
 - Set IP Range End to appropriate value (default 192.168.1.254)
 - Set vlan to appropriate value (default 102)
5. Update the Management network configuration.
 - Set CIDR to appropriate value (default 192.168.0.0/24)
 - Set IP Range Start to appropriate value (default 192.168.0.1)
 - Set IP Range End to appropriate value (default 192.168.0.254)
 - Check <VLAN tagging>.
 - Set appropriate VLAN id. (default 101)
6. Update the Private Network Information
 - Set CIDR to appropriate value (default 192.168.2.0/24)
 - Set IP Range Start to appropriate value (default 192.168.2.1)
 - Set IP Range End to appropriate value (default 192.168.2.254)
 - Check <VLAN tagging>.
 - Set appropriate VLAN tag (default 103)
7. Select the “Neutron L3 Node Networks group” on the left pane.

8. Update the Floating Network configuration.
 - Set the Floating IP range start (default 172.16.0.130)
 - Set the Floating IP range end (default 172.16.0.254)
 - Set the Floating network name (default admin_floating_net)
9. Update the Internal Network configuration.
 - Set Internal network CIDR to an appropriate value (default 192.168.111.0/24)
 - Set Internal network gateway to an appropriate value
 - Set the Internal network name (default admin_internal_net)
10. Update the Guest OS DNS servers.
 - Set Guest OS DNS Server values appropriately
11. Save Settings.
12. Select the “Other Node Networks group” on the left pane.
13. Update the Public network assignment.
 - Check the box for “Assign public network to all nodes” (Required by OpenDaylight)
14. Update Host OS DNS Servers.
 - Provide the DNS server settings
15. Update Host OS NTP Servers.
 - Provide the NTP server settings

SELECT HYPERVISOR TYPE

1. In the FUEL UI of your Environment, click the “Settings” Tab
2. Select Compute on the left side pane
 - Check the KVM box and press “Save settings”

ENABLE PLUGINS

1. In the FUEL UI of your Environment, click the “Settings” Tab
2. Select Other on the left side pane
 - Enable and configure the plugins of your choice

ALLOCATE NODES TO ENVIRONMENT AND ASSIGN FUNCTIONAL ROLES

1. Click on the “Nodes” Tab in the FUEL WEB UI.
2. Assign roles.
 - Click on the <+Add Nodes> button
 - Check <Controller>, <Telemetry - MongoDB> and optionally an SDN Controller role (OpenDaylight controller/ONOS) in the Assign Roles Section.
 - Check one node which you want to act as a Controller from the bottom half of the screen
 - Click <Apply Changes>.
 - Click on the <+Add Nodes> button
 - Check the <Controller> and <Storage - Ceph OSD> roles.
 - Check the two next nodes you want to act as Controllers from the bottom half of the screen
 - Click <Apply Changes>
 - Click on <+Add Nodes> button
 - Check the <Compute> and <Storage - Ceph OSD> roles.
 - Check the Nodes you want to act as Computes from the bottom half of the screen
 - Click <Apply Changes>.
3. Configure interfaces.
 - Check Select <All> to select all allocated nodes
 - Click <Configure Interfaces>
 - Assign interfaces (bonded) for mgmt-, admin-, private-, public- and storage networks
 - Click <Apply>

OPTIONAL - SET LOCAL MIRROR REPOS

The following steps can be executed if you are in an environment with no connection to the Internet. The Fuel server delivers a local repo that can be used for installation / deployment of openstack.

1. In the Fuel UI of your Environment, click the Settings Tab and select General from the left pane.
 - Replace the URI values for the “Name” values outlined below:
 - “ubuntu” URI=”deb <http://<ip-of-fuel-server>:8080/mirrors/ubuntu/> trusty main”
 - “ubuntu-security” URI=”deb <http://<ip-of-fuel-server>:8080/mirrors/ubuntu/> trusty-security main”
 - “ubuntu-updates” URI=”deb <http://<ip-of-fuel-server>:8080/mirrors/ubuntu/> trusty-updates main”
 - “mos” URI=”deb http://<ip-of-fuel-server>:8080/liberty-8.0/ubuntu/x86_64 mos8.0 main restricted”
 - “Auxiliary” URI=”deb <http://<ip-of-fuel-server>:8080/liberty-8.0/ubuntu/auxiliary> auxiliary main restricted”
 - Click <Save Settings> at the bottom to Save your changes

TARGET SPECIFIC CONFIGURATION

1. Set up targets for provisioning with non-default “Offloading Modes”

Some target nodes may require additional configuration after they are PXE booted (bootstrapped); the most frequent changes are in defaults for ethernet devices’ “Offloading Modes” settings (e.g. some targets’ ethernet drivers may strip VLAN traffic by default).

If your target ethernet drivers have wrong “Offloading Modes” defaults, in “Configure interfaces” page (described above), expand affected interface’s “Offloading Modes” and [un]check the relevant settings:

2. Set up targets for “Verify Networks” with non-default “Offloading Modes”

NOTE: Check *Reference 15* for an updated and comprehensive list of known issues and/or limitations, including “Offloading Modes” not being applied during “Verify Networks” step.

Setting custom “Offloading Modes” in Fuel GUI will only apply those settings during provisioning and **not** during “Verify Networks”, so if your targets need this change, you have to apply “Offloading Modes” settings by hand to bootstrapped nodes.

E.g.: Our driver has “rx-vlan-filter” default “on” (expected “off”) on the openstack interface(s) “eth1”, preventing VLAN traffic from passing during “Verify Networks”.

- From Fuel master console identify target nodes admin IPs:

```
$ fuel nodes
```

- SSH into each of the target nodes and disable “rx-vlan-filter” on the affected physical interface(s) allocated for OpenStack traffic (eth1):

```
$ ssh root@10.20.0.6 ethtool -K eth1 rx-vlan-filter off
```

- Repeat the step above for all affected nodes/interfaces in the POD.

VERIFY NETWORKS

It is important that the Verify Networks action is performed as it will verify that communication works for the networks you have setup, as well as check that packages needed for a successful deployment can be fetched.

From the FUEL UI in your Environment, Select the Networks Tab and select “Connectivity check” on the left pane

- Select <Verify Networks>
- Continue to fix your topology (physical switch, etc) until the “Verification Succeeded” and “Your network is configured correctly” message is shown

DEPLOY YOUR ENVIRONMENT

Deploy the environment.

- In the Fuel GUI, click on the “Dashboard” Tab.
- Click on <Deploy Changes> in the “Ready to Deploy?” section
- Examine any information notice that pops up and click <Deploy>

Wait for your deployment to complete, you can view the “Dashboard” Tab to see the progress and status of your deployment.

INSTALLATION HEALTH-CHECK

1. Perform system health-check (see figure below)
 - Click the “Health Check” tab inside your Environment in the FUEL Web UI
 - Check <Select All> and Click <Run Tests>
 - Allow tests to run and investigate results where appropriate

REFERENCES

15.1 OPNFV

1. OPNFV Home Page
2. OPNFV documentation- and software downloads

15.2 OpenStack

3. OpenStack Liberty Release artifacts
4. OpenStack documentation

15.3 OpenDaylight

5. OpenDaylight artifacts

15.4 Fuel

6. The Fuel OpenStack project
7. Fuel documentation overview
8. Fuel planning guide
9. Fuel quick start guide
10. Fuel operations guide
11. Fuel Plugin Developers Guide
12. Fuel OpenStack Hardware Compatibility List

15.5 Fuel in OPNFV

13. OPNFV Installation instruction for the Brahmaputra release of OPNFV when using Fuel as a deployment tool
14. OPNFV Build instruction for the Brahmaputra release of OPNFV when using Fuel as a deployment tool

15. [OPNFV Release Note for the Brahmaputra release of OPNFV when using Fuel as a deployment tool](#)