Fuel@OPNFV Installation Instruction

Release draft (34a852a)

OPNFV

August 18, 2016

CHAPTER

ONE

OPNFV INSTALLATION INSTRUCTION FOR THE COLORADO RELEASE OF OPNFV WHEN USING FUEL AS A DEPLOYMENT TOOL

1.1 License

1.2 Abstract

This document describes how to install the Colorado release of OPNFV when using Fuel as a deployment tool, covering it's usage, limitations, dependencies and required system resources.

1.3 Introduction

This document provides guidelines on how to install and configure the Colorado release of OPNFV when using Fuel as a deployment tool, 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 Colorado compliant deployment.

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

1.4 Preface

Before starting the installation of the Colorado release of OPNFV, using Fuel as a deployment tool, some planning must be done.

1.4.1 Retrieving the ISO image

First of all, the Fuel deployment ISO image needs to be retrieved, the Fuel .iso image of the Colorado release can be found at *Reference: 2*

1.4.2 Building the ISO image

Alternatively, you may 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*

1.4.3 Other preparations

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

- Fuel Installation Guide, please see *Reference: 8*
- Fuel User Guide, please see *Reference: 9*
- Fuel Developer 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, erc.).
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.

1.5 Hardware requirements

The following minimum hardware requirements must be met for the installation of Colorado using Fuel:

HW Aspect	Requirement
# of nodes	Minimum 5 (3 for non redundant deployment): • 1 Fuel deployment master (may be virtualized) • 3(1) Controllers (1 colocated mongo/ceilometer role, 2 Ceph-OSD roles) • 1 Compute (1 co-located Ceph-OSD role)
CPU	Minimum 1 socket x86_AMD64 with Virtualization support
RAM	Minimum 16GB/server (Depending on VNF work load)
Disk	Minimum 256GB 10kRPM spinning disks
Networks	4 Tagged VLANs (PUBLIC, MGMT, STORAGE, PRIVATE) 1 Un-Tagged VLAN for PXE Boot - ADMIN Network Note: These can be allocated to a single NIC - or spread out over multiple NICs as your hardware supports.

1.6 Help with Hardware Requirements

Calculate hardware requirements:

For information on compatible hardware types available for use, please see *Reference: 11*.

When choosing the hardware on which you will deploy your OpenStack environment, you should think about:

- CPU – Consider the number of virtual machines that you plan to deploy in your cloud environment and the CPU per virtual machine.
- Memory – Depends on the amount of RAM assigned per virtual machine and the controller node.
- Storage – Depends on the local drive space per virtual machine, remote volumes that can be attached to a virtual machine, and object storage.
- Networking – Depends on the Choose Network Topology, the network bandwidth per virtual machine, and network storage.

1.7 Top of the rack (TOR) Configuration requirements

The switching infrastructure provides connectivity for the OPNFV infrastructure operations, tenant networks (East/West) and provider connectivity (North/South); it also provides needed connectivity for the Storage Area Network (SAN). To avoid traffic congestion, it is strongly suggested that three physically separated networks are used, that is: 1 physical network for administration and control, one physical network for tenant private and public networks, and one physical network for SAN. The switching connectivity can (but does not need to) be fully redundant, in such case it comprises a redundant 10GE switch pair for each of the three physically separated networks.

The physical TOR switches are **not** automatically configured from the Fuel OPNFV reference platform. All the networks involved in the OPNFV infrastructure as well as the provider networks and the private tenant VLANs needs to be manually configured.

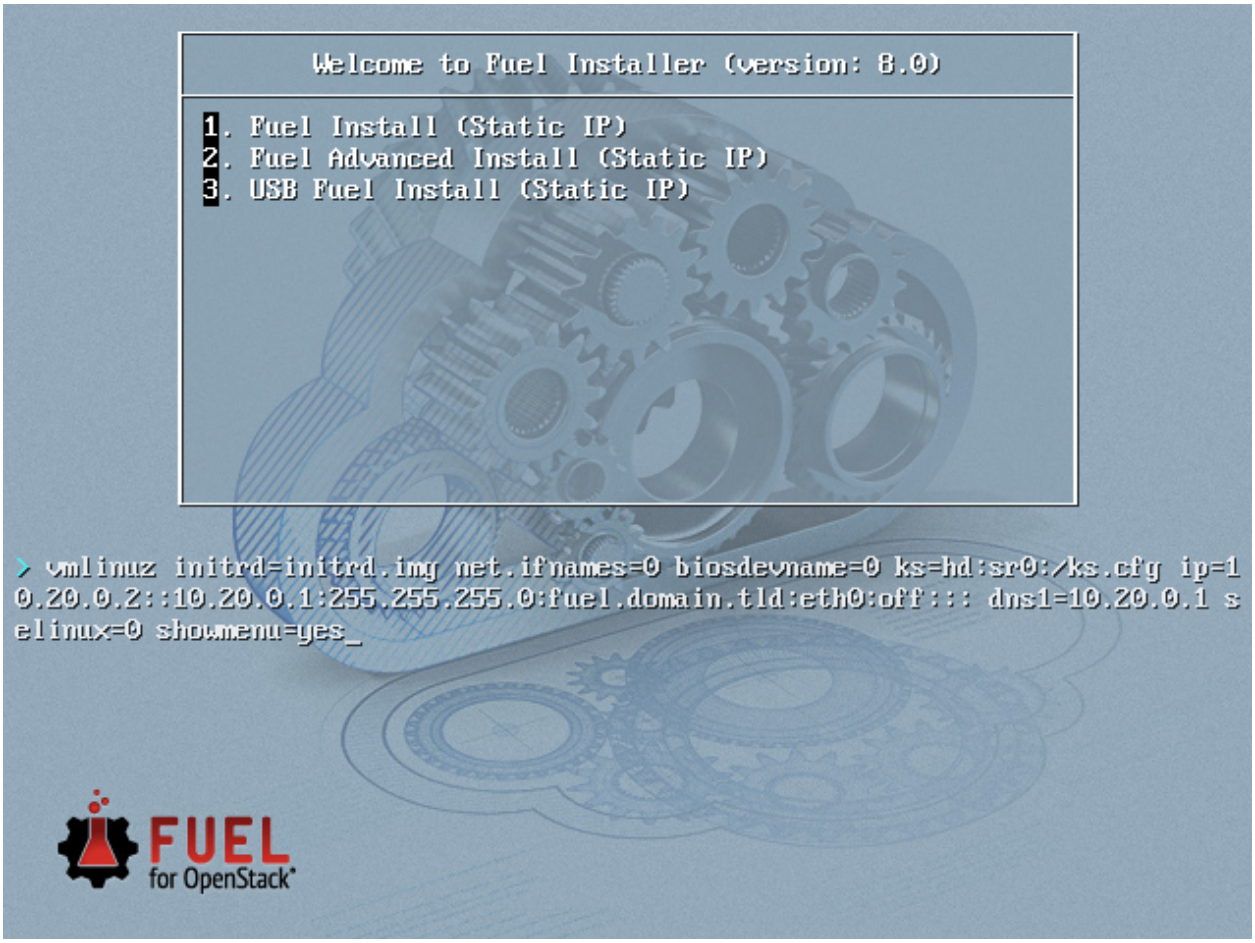
Manual configuration of the Colorado hardware platform should be carried out according to the OPNFV Pharos specification: <https://wiki.opnfv.org/pharos/pharos_specification>

1.8 OPNFV Software installation and deployment

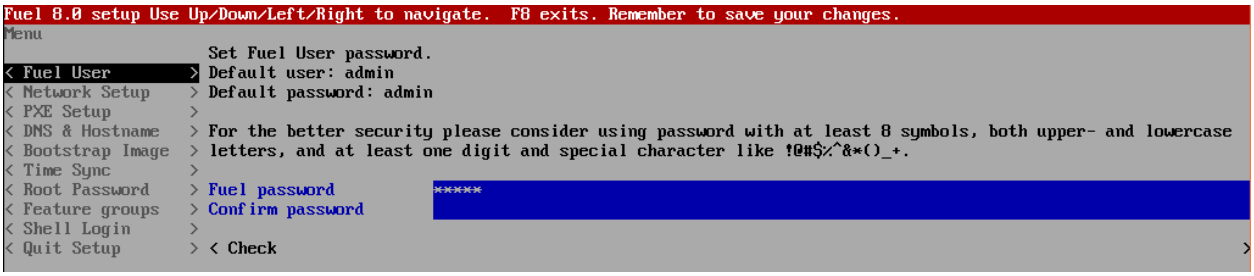
This section describes the installation of the OPNFV installation server (Fuel master) as well as the deployment of the full OPNFV reference platform stack across a server cluster.

1.8.1 Install Fuel master

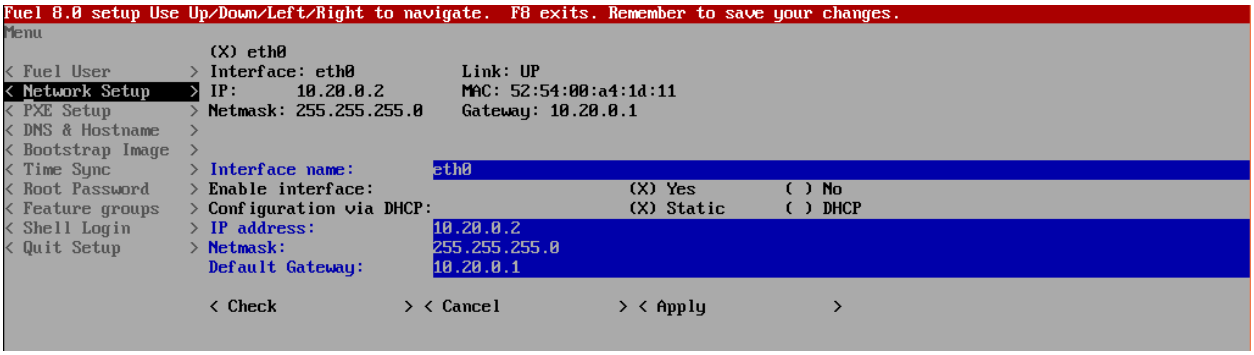
1. Mount the Colorado Fuel ISO file/media as a boot device to the jump host server.
2. Reboot the jump host to establish the Fuel server.
   - The system now boots from the ISO image.
   - Select "Fuel Install (Static IP)" (See figure below)
   - Press [Enter].



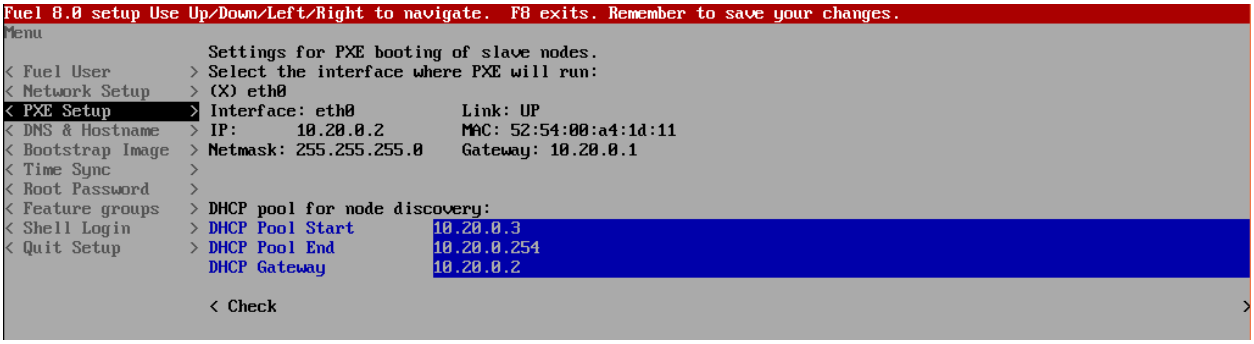
3. Wait until screen Fuel setup is shown (Note: This can take up to 30 minutes).
4. In the "Fuel User" section - Confirm/change the default password (See figure below)
   - Enter "admin" in the Fuel password input
   - Enter "admin" in the Confirm password input
   - Select "Check" and press [Enter]



5. 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 (see figure below).
    - Configure eth1 or other network interfaces here as well (if you have them present on your FUEL server).

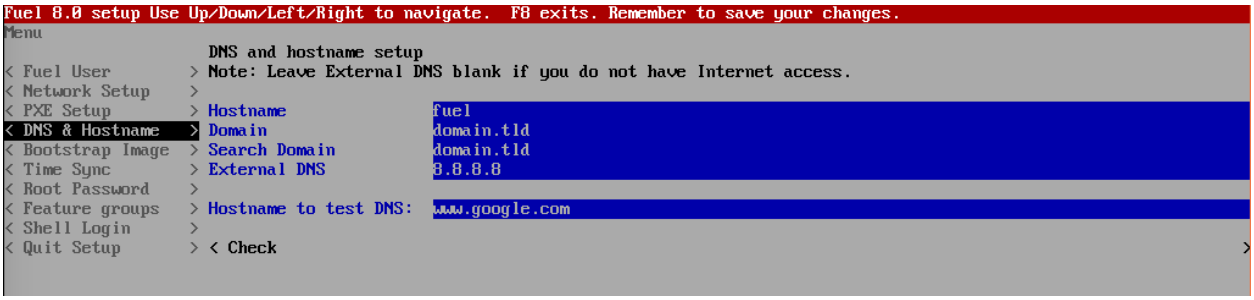


6. In the “PXE Setup” section (see figure below) - Change the following fields to appropriate values (example below):
    - DHCP Pool Start 10.20.0.3
    - DHCP Pool End 10.20.0.254
    - DHCP Pool Gateway 10.20.0.2 (IP address of Fuel node)

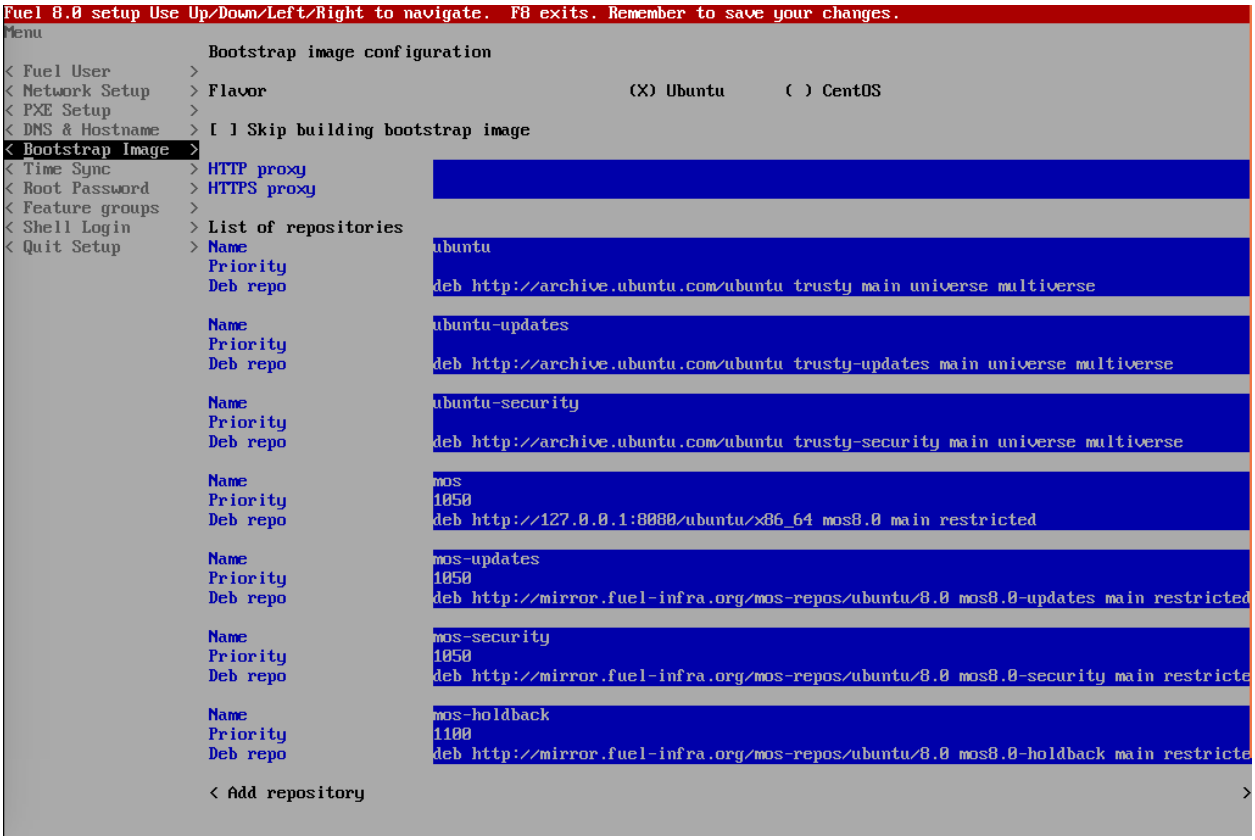


7. In the “DNS & Hostname” section (see figure below) - Change the following fields to appropriate values:
    - Hostname
    - Domain
    - Search Domain
    - External DNS
    - Hostname to test DNS

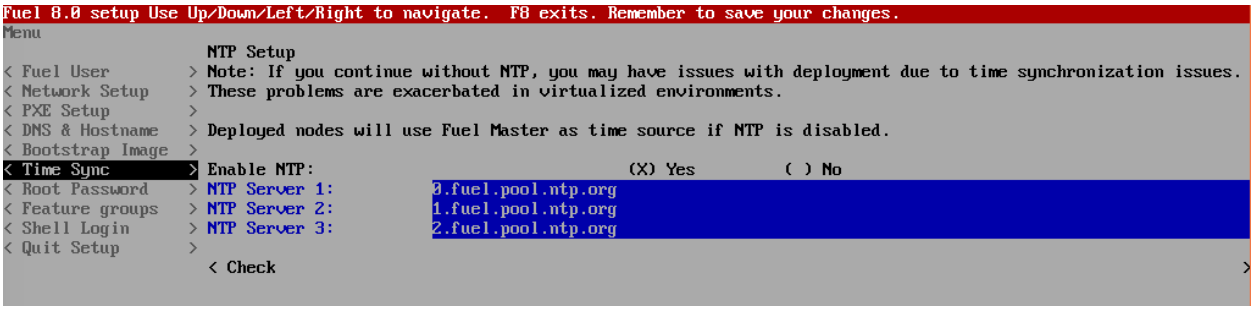
- Select <Check> and press [Enter]



8. OPTION TO ENABLE PROXY SUPPORT - In the “Bootstrap Image” section (see figure below), 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]



9. In the “Time Sync” section (see figure below) - 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>
10. Start the installation.
   - Select Quit Setup and press Save and Quit.

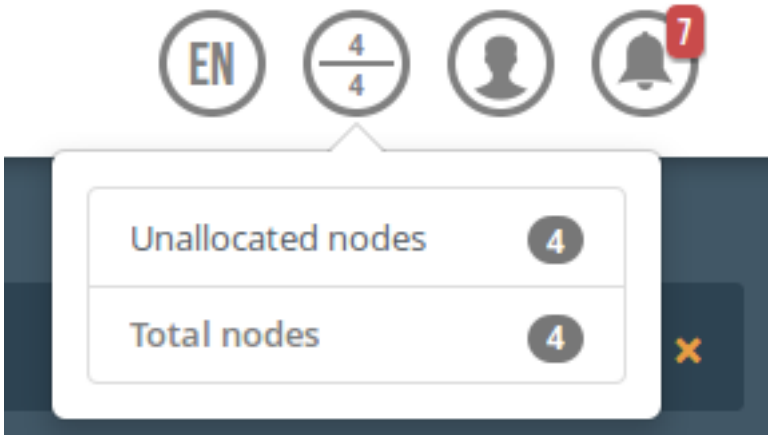


- Installation starts, wait until the login screen is shown.

1.8.2 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 (see figure below).



1.8.3 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
```

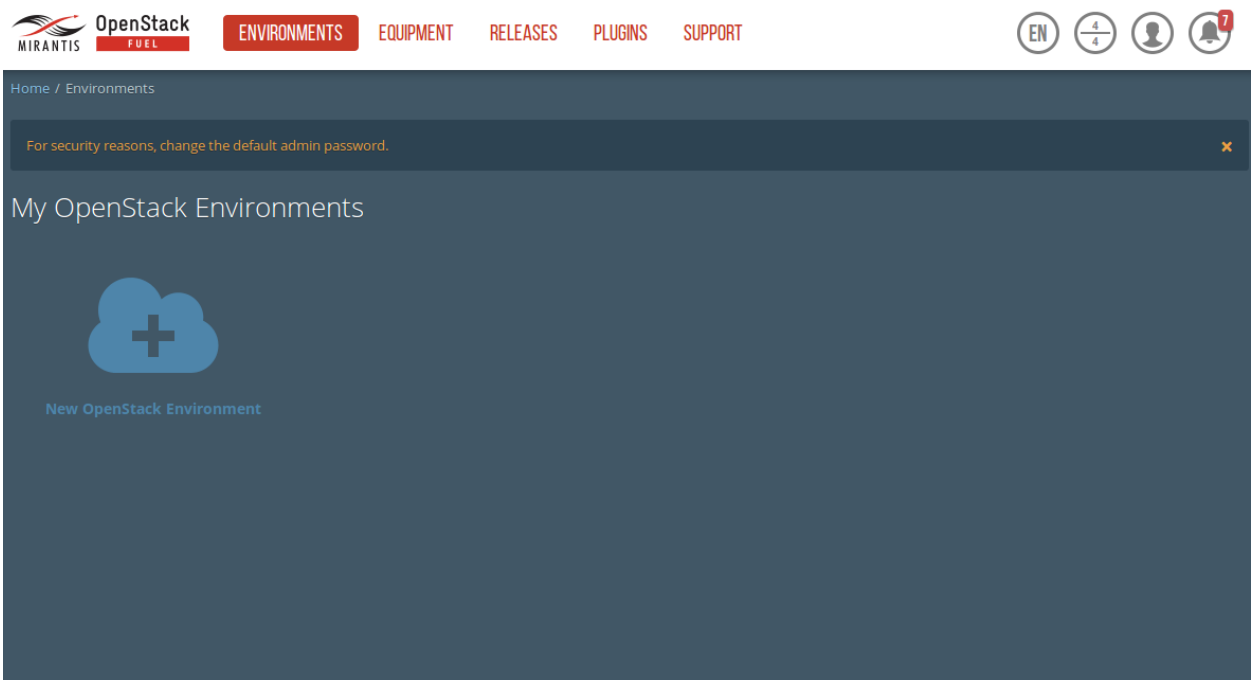
Expected output (see figure below):

```
Plugin ..... was successfully installed.
```

```
[root@fuel opnfv]# pwd
/opt/opnfv
[root@fuel opnfv]# ls
bootstrap                           fuel-plugin-qemu-0.5-0.5.2-1.noarch.rpm    onos-0.8-0.8.0-1.noarch.rpm
fuel-plugin-ovs-0.5-0.5.2-1.noarch.rpm  fuel-plugin-vsperf-1.0-1.0.0-1.noarch.rpm  opendaylight-0.8-0.8.0-1.noarch.rpm
[root@fuel opnfv]# fuel plugins --install opendaylight-0.8-0.8.0-1.noarch.rpm
Loaded plugins: fastestmirror, priorities
Examining opendaylight-0.8-0.8.0-1.noarch.rpm: opendaylight-0.8-0.8.0-1.noarch
Marking opendaylight-0.8-0.8.0-1.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package opendaylight-0.8.noarch 0:0.8.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                      Arch                Version              Repository                                      Size
================================================================================================================================
Installing:
 opendaylight-0.8             noarch              0.8.0-1              /opendaylight-0.8-0.8.0-1.noarch               282 M

Transaction Summary
================================================================================================================================
Install  1 Package

Total size: 282 M
Installed size: 282 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : opendaylight-0.8-0.8.0-1.noarch                                                                    1/1
  Verifying  : opendaylight-0.8-0.8.0-1.noarch                                                                    1/1

Installed:
  opendaylight-0.8.noarch 0:0.8.0-1

Complete!
Plugin opendaylight-0.8-0.8.0-1.noarch.rpm was successfully installed.
[root@fuel opnfv]#
```

1.8.4 Create an OpenStack Environment

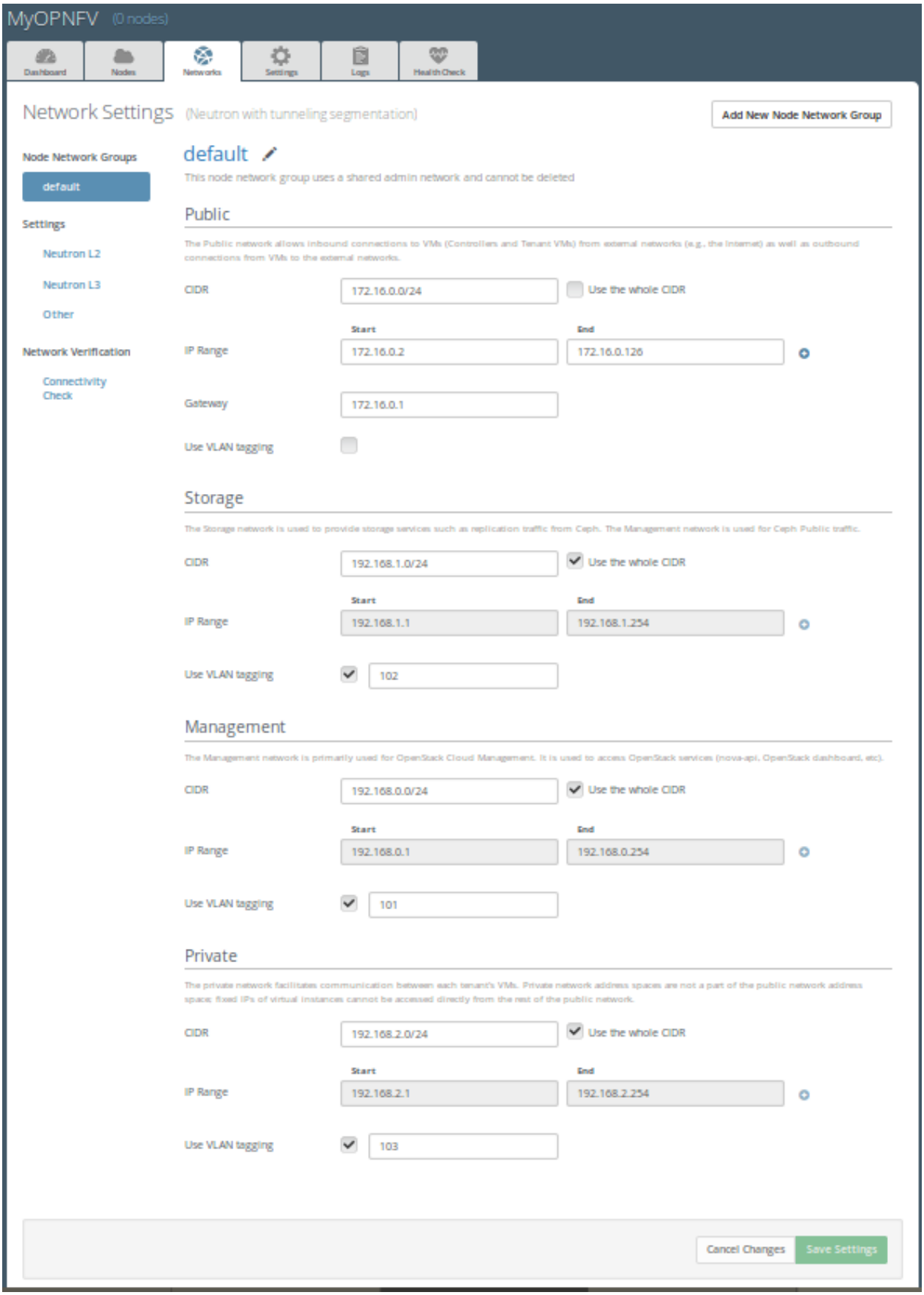
1. Connect to Fuel WEB UI with a browser (default: https://10.20.0.2:8443) (login admin/admin)
2. Create and name a new OpenStack environment, to be installed.
3. Select “<Mitaka on Ubuntu 14.04>” and press <Next>
4. Select “compute virtulization method”.
   - Select “QEMU-KVM as hypervisor” and press <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>



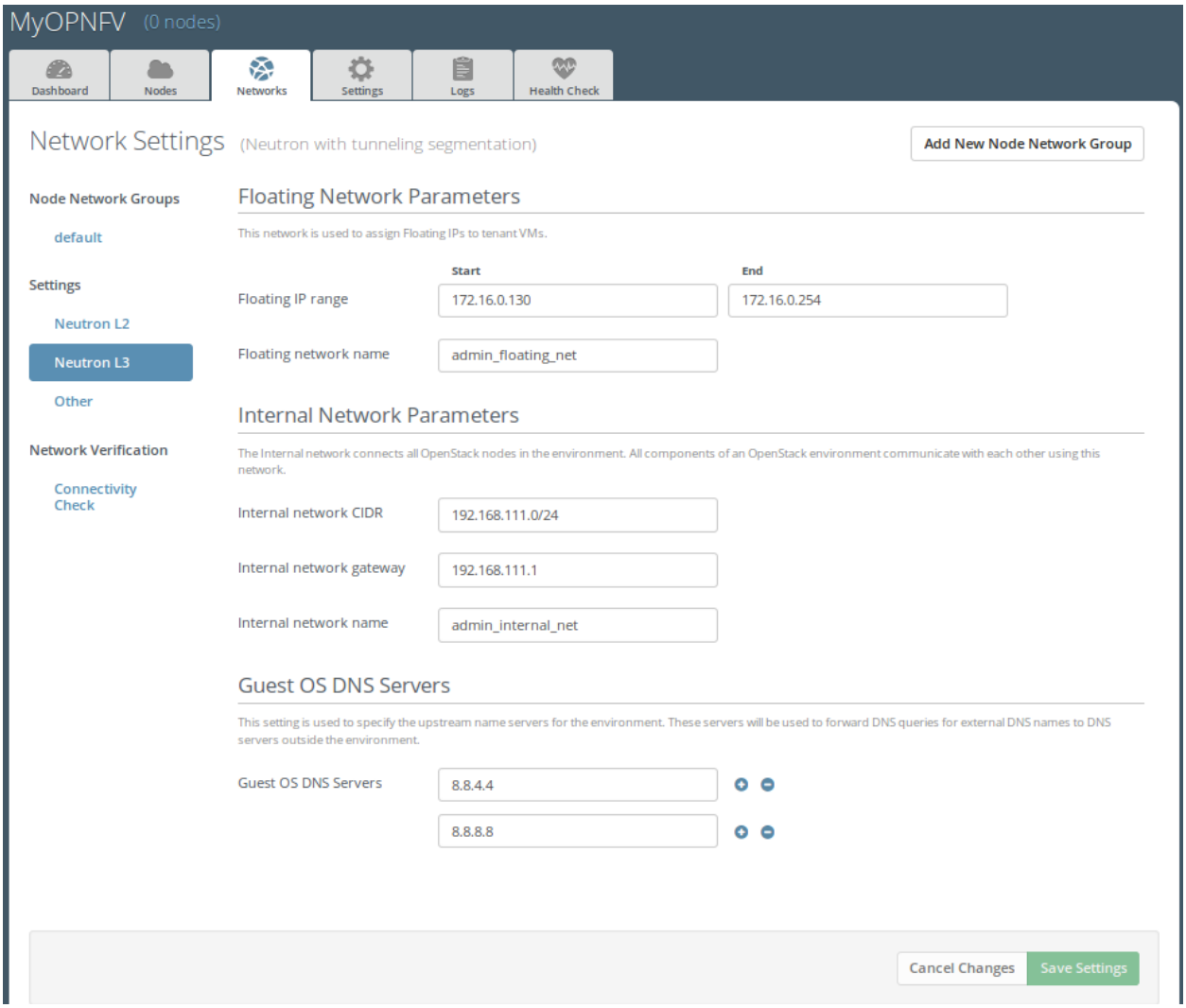
7. Select “additional services” you wish to install.
    - Check option “Install Ceilometer (OpenStack Telemetry)” and press <Next>
8. Create the new environment.
    - Click <Create> Button

1.8.5 Configure the network environment

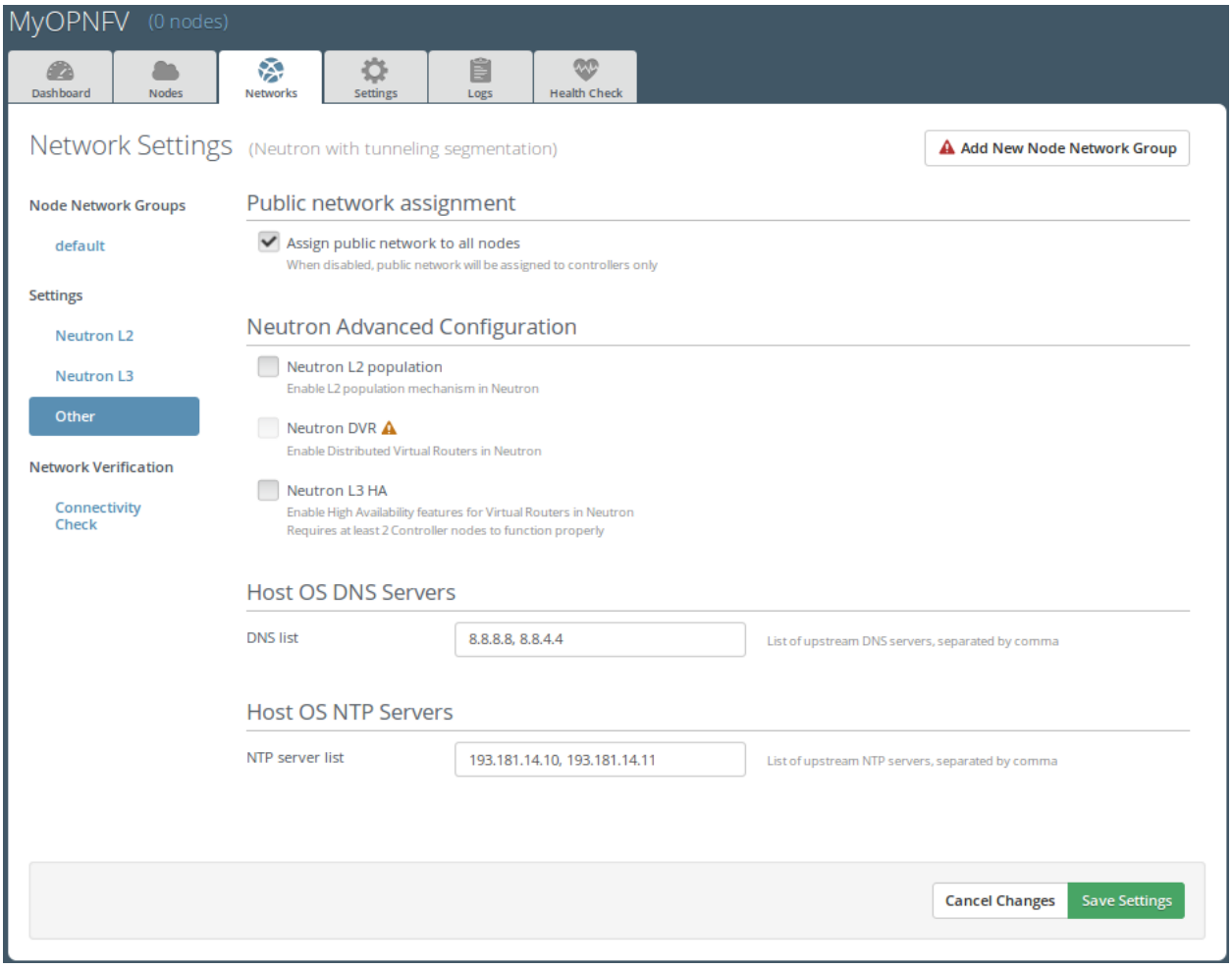
1. Open the environment you previously created.
2. Open the networks tab and select the “default Node Networks group to” on the left pane (see figure below).
3. Update the Public network configuration and change the following fields to appropriate values:
    - CIDR to <CIDR for Public IP Addresses>
    - 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)


MyOPNFV (0 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
Network Settings (Neutron with tunneling segmentation)
Add New Node Network Group
Node Network Groups
default
Settings
Neutron L2
Neutron L3
Other
Network Verification
Connectivity Check
default
This node network group uses a shared admin network and cannot be deleted
Public
The Public network allows inbound connections to VMs (Controllers and Tenant VMs) from external networks (e.g., the Internet) as well as outbound connections from VMs to the external networks.
CIDR
172.16.0.0/24
Use the whole CIDR
Start
End
IP Range
172.16.0.2
172.16.0.126
Gateway
172.16.0.1
Use VLAN tagging
Storage
The Storage network is used to provide storage services such as replication traffic from Ceph. The Management network is used for Ceph Public traffic.
CIDR
192.168.1.0/24
Use the whole CIDR
Start
End
IP Range
192.168.1.1
192.168.1.254
Use VLAN tagging
102
Management
The Management network is primarily used for OpenStack Cloud Management. It is used to access OpenStack services (nova-api, OpenStack dashboard, etc).
CIDR
192.168.0.0/24
Use the whole CIDR
Start
End
IP Range
192.168.0.1
192.168.0.254
Use VLAN tagging
101
Private
The private network facilitates communication between each tenant's VMs. Private network address spaces are not a part of the public network address space; fixed IPs of virtual instances cannot be accessed directly from the rest of the public network.
CIDR
192.168.2.0/24
Use the whole CIDR
Start
End
IP Range
192.168.2.1
192.168.2.254
Use VLAN tagging
103
Cancel Changes
Save Settings

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(see figure below).

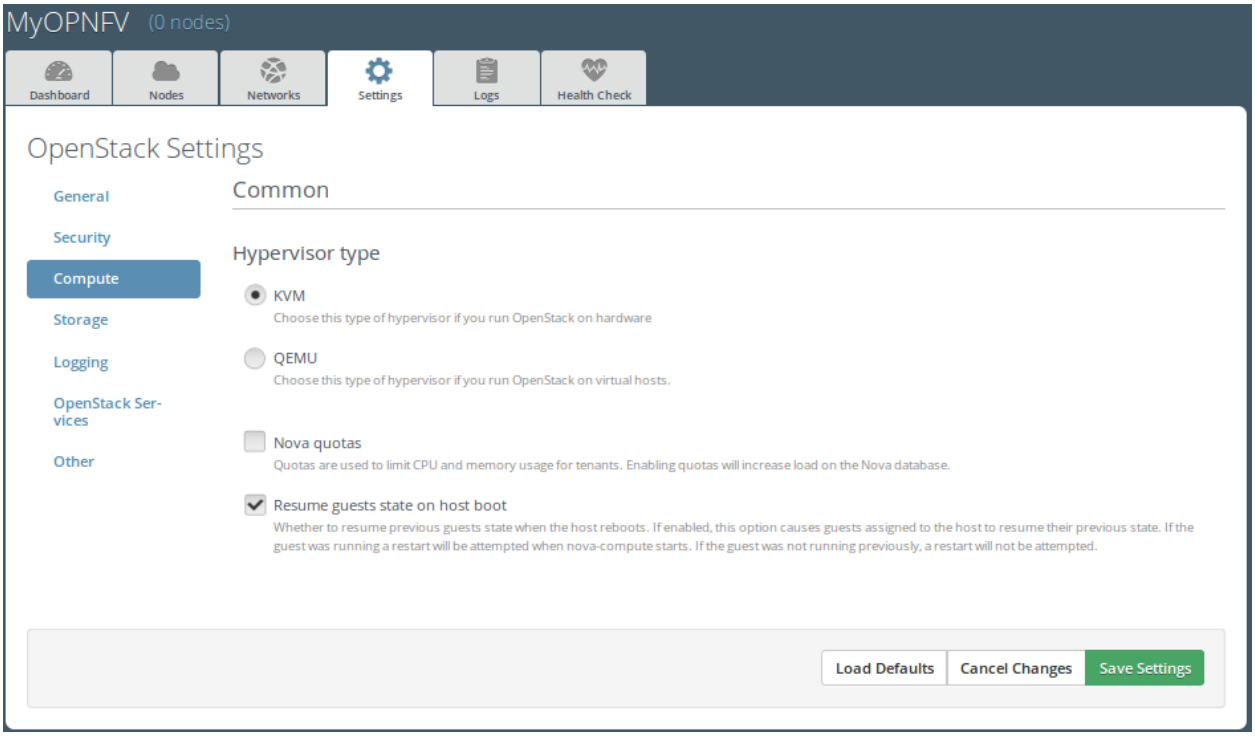


13. Update the Public network assignment.
    - Check the box for "Assign public network to all nodes" (Required by OpenDaylight)

14. Update Host OS DNS Servers.
    - Provide the DNS server settings
15. Update Host OS NTP Servers.
    - Provide the NTP server settings

1.8.6 Select Hypervisor type

1. In the FUEL UI of your Environment, click the "Settings" Tab
2. Select Compute on the left side pane (see figure below)
    - Check the KVM box and press "Save settings"



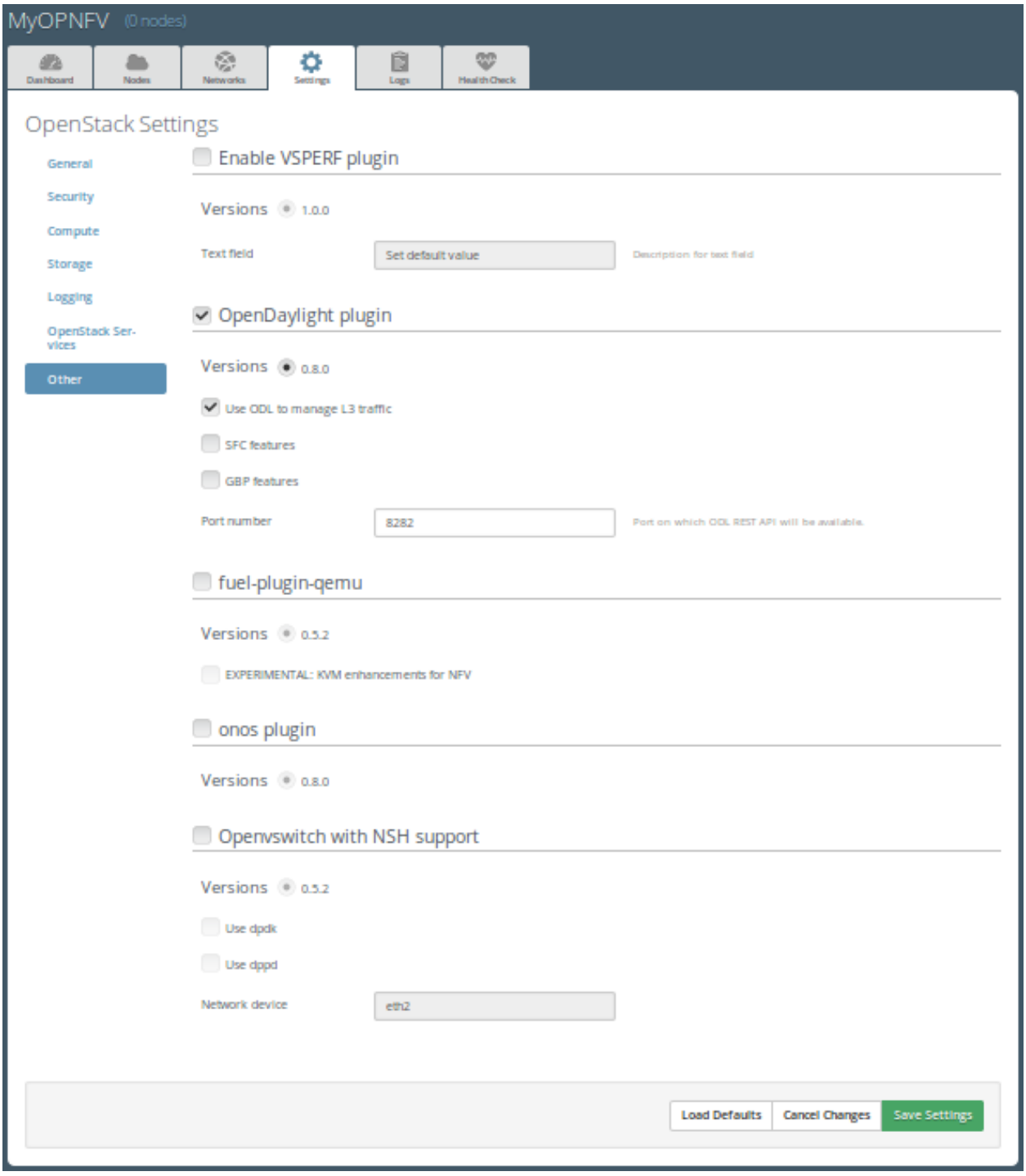
1.8.7 Enable Plugins

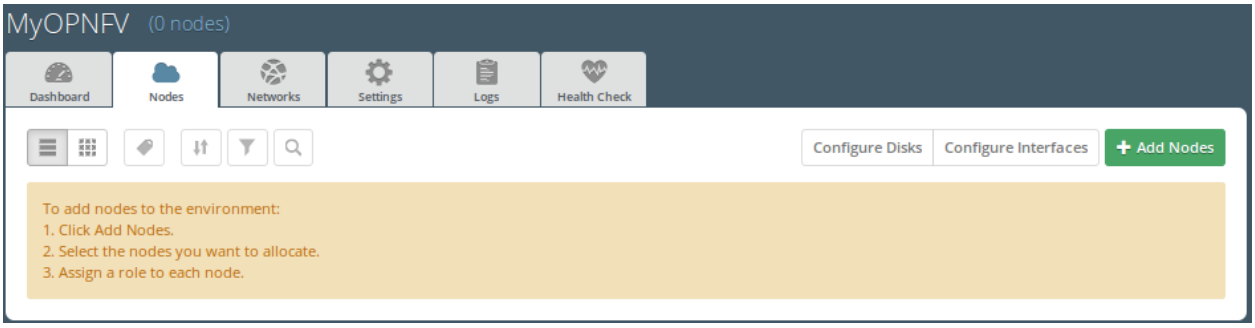
1. In the FUEL UI of your Environment, click the "Settings" Tab
2. Select Other on the left side pane (see figure below)
    - Enable and configure the plugins of your choice

1.8.8 Allocate nodes to environment and assign functional roles

1. Click on the "Nodes" Tab in the FUEL WEB UI (see figure below).

2. Assign roles (see figure below).


MyOPNFV (0 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
OpenStack Settings
General
Security
Compute
Storage
Logging
OpenStack Services
Other
Enable VSPERF plugin
Versions 1.0.0
Text field
Set default value
Description for text field
OpenDaylight plugin
Versions 0.8.0
Use ODL to manage L3 traffic
SFC features
GBP features
Port number
8282
Port on which ODL REST API will be available.
fuel-plugin-qemu
Versions 0.5.2
EXPERIMENTAL: KVM enhancements for NFV
onos plugin
Versions 0.8.0
Openvswitch with NSH support
Versions 0.5.2
Use dpdk
Use dppd
Network device
eth2
Load Defaults
Cancel Changes
Save Settings



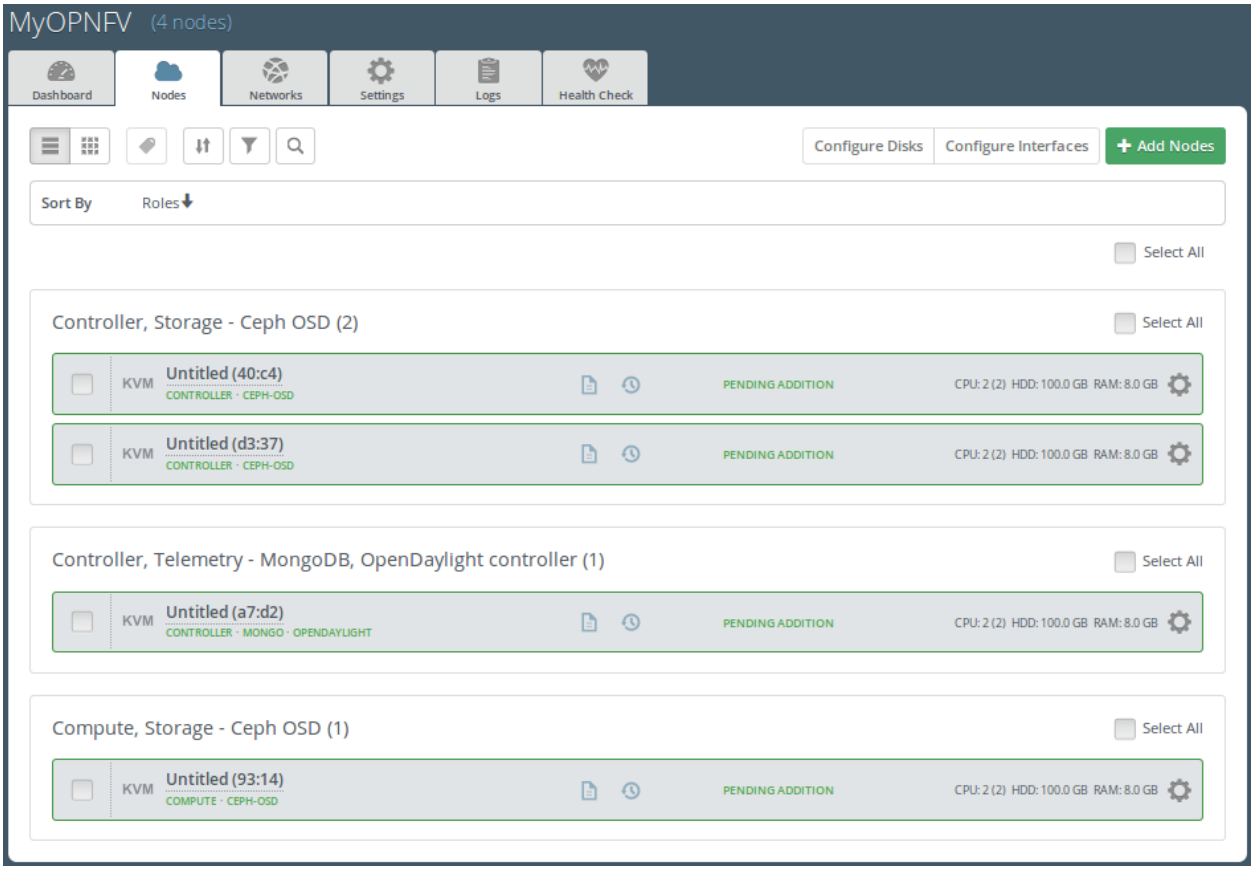
- Click on the <+Add Nodes> button
- Check <Controller>, <Telemetry - MongoDB> and optionally an SDN Controller role (OpenDaylight controller/ONOS) in the Assign Roles Section.
- Check one node which you want to act as a Controller from the bottom half of the screen
- Click <Apply Changes>.
- Click on 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 (see figure below).
    - Check Select <All> to select all allocated nodes
    - Click <Configure Interfaces>
    - Assign interfaces (bonded) for mgmt-, admin-, private-, public- and storage networks
    - Click <Apply>

1.8.9 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/mitaka-9.0/ubuntu/x86_64 mos8.0 main restricted"



- "Auxiliary" URI="deb http://<ip-of-fuel-server>:8080/mitaka-9.0/ubuntu/auxiliary auxiliary main restricted"
- Click <Save Settings> at the bottom to Save your changes

1.8.10 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).

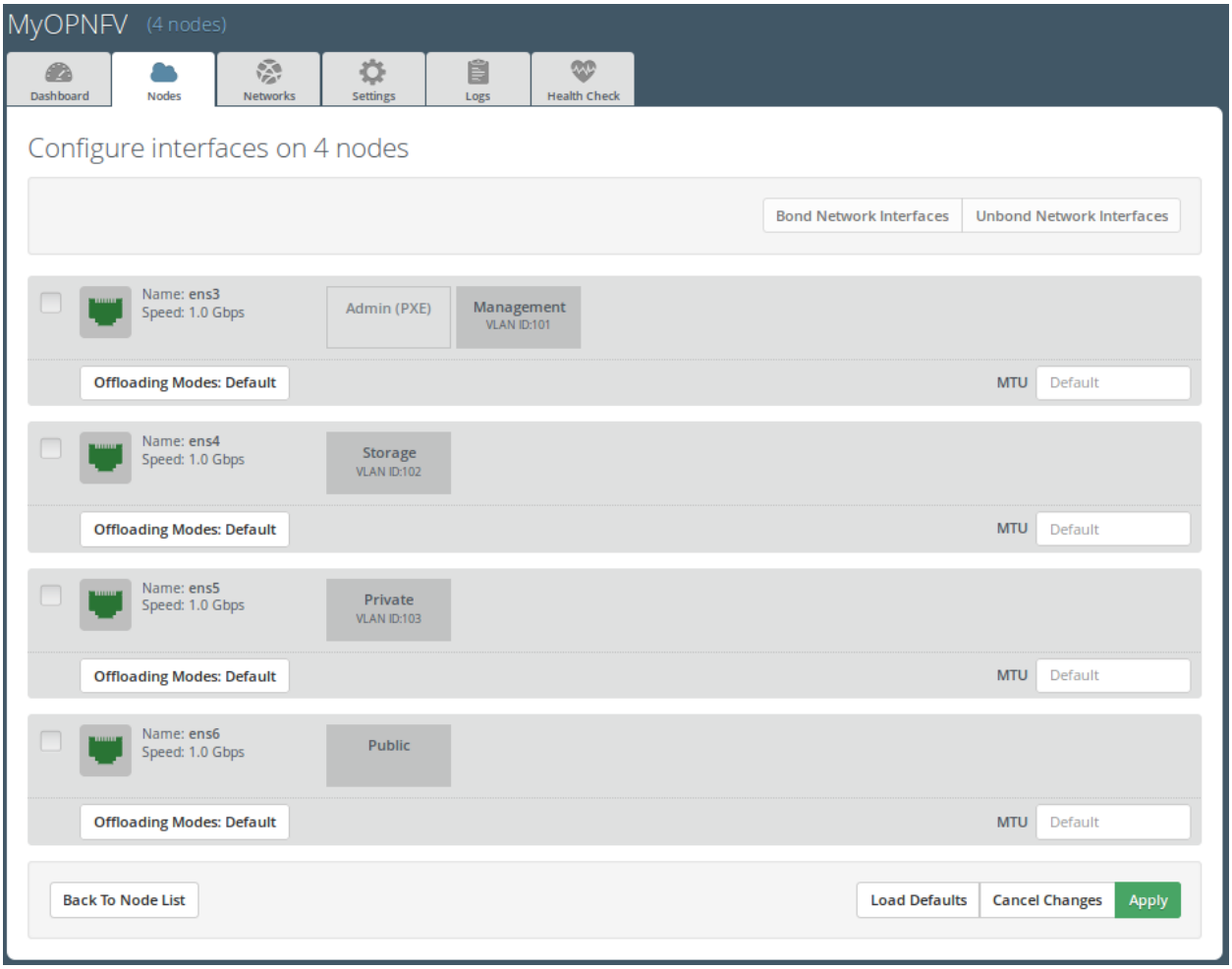
   If your target ethernet drivers have wrong "Offloading Modes" defaults, in "Configure interfaces" page (described above), expand affected interface's "Offloading Modes" and [un]check the relevant settings (see figure below):

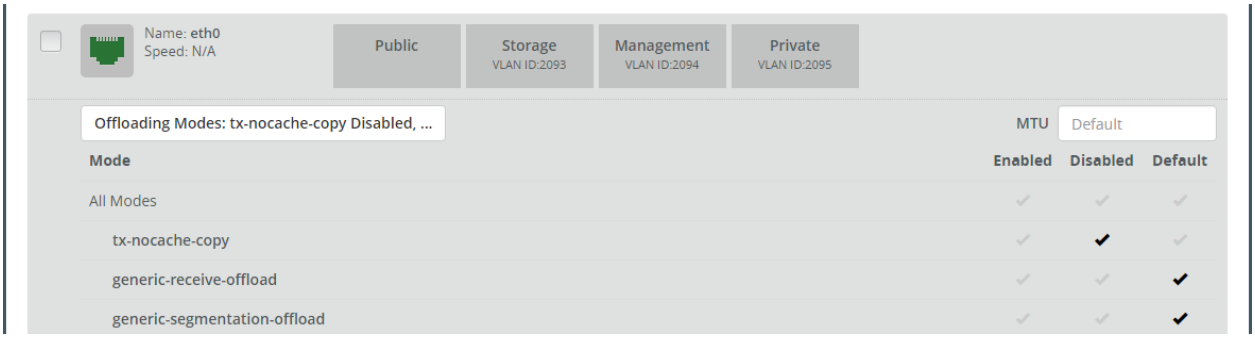
2. Set up targets for "Verify Networks" with non-default "Offloading Modes"

   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 provisiong 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".


MyOPNFV (4 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
Configure interfaces on 4 nodes
Bond Network Interfaces
Unbond Network Interfaces
Name: ens3
Speed: 1.0 Gbps
Admin (PXE)
Management
VLAN ID:101
Offloading Modes: Default
MTU
Default
Name: ens4
Speed: 1.0 Gbps
Storage
VLAN ID:102
Offloading Modes: Default
MTU
Default
Name: ens5
Speed: 1.0 Gbps
Private
VLAN ID:103
Offloading Modes: Default
MTU
Default
Name: ens6
Speed: 1.0 Gbps
Public
Offloading Modes: Default
MTU
Default
Back To Node List
Load Defaults
Cancel Changes
Apply


Name: eth0
Speed: N/A
Public
Storage
VLAN ID:2093
Management
VLAN ID:2094
Private
VLAN ID:2095
Offloading Modes: tx-nocache-copy Disabled, ...
MTU
Default
Mode
Enabled
Disabled
Default
All Modes
tx-nocache-copy
generic-receive-offload
generic-segmentation-offload

- From Fuel master console identify target nodes admin IPs (see figure below):

```
$ fuel nodes
```

```
[root@fuel ~]# fuel nodes
id | status   | name              | cluster | ip         | mac               | roles                          | pending_roles | online | group_id
---|----------|-------------------|---------|------------|-------------------|--------------------------------|---------------|--------|---------
3  | ready    | softiron-1 (05:96) | 1       | 10.20.0.7  | e0:ff:f7:00:05:96 | cinder, compute                |               | True   | 1
2  | ready    | softiron-2 (05:93) | 1       | 10.20.0.6  | e0:ff:f7:00:05:93 | cinder, controller, opendaylight |             | True   | 1
```

- 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.

1.8.11 Verify Networks

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

1. From the FUEL UI in your Environment, Select the Networks Tab and select "Connectivity check" on the left pane (see figure below)
   - Select <Verify Networks>
   - Continue to fix your topology (physical switch, etc) until the "Verification Succeeded" and "Your network is configured correctly" message is shown

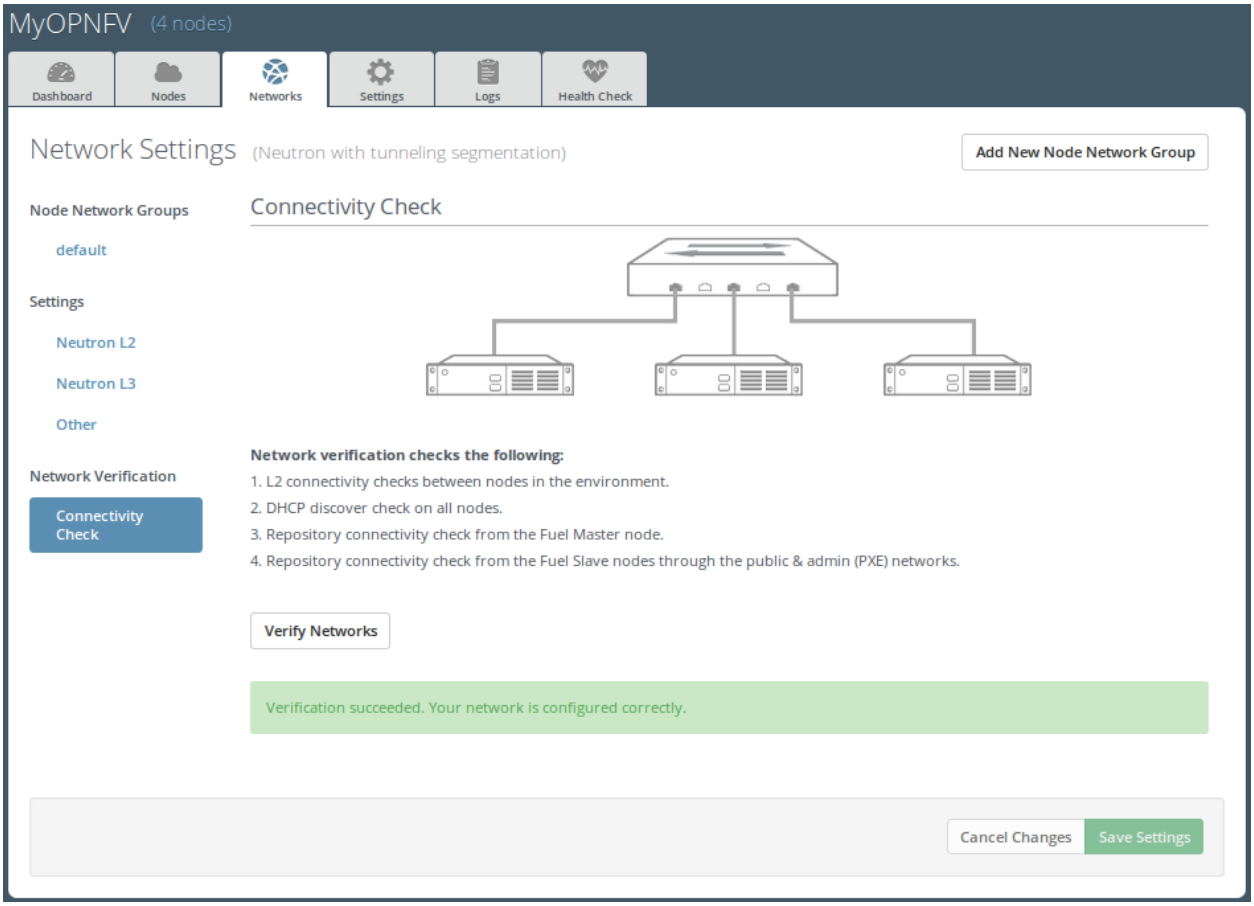
1.8.12 Deploy Your Environment

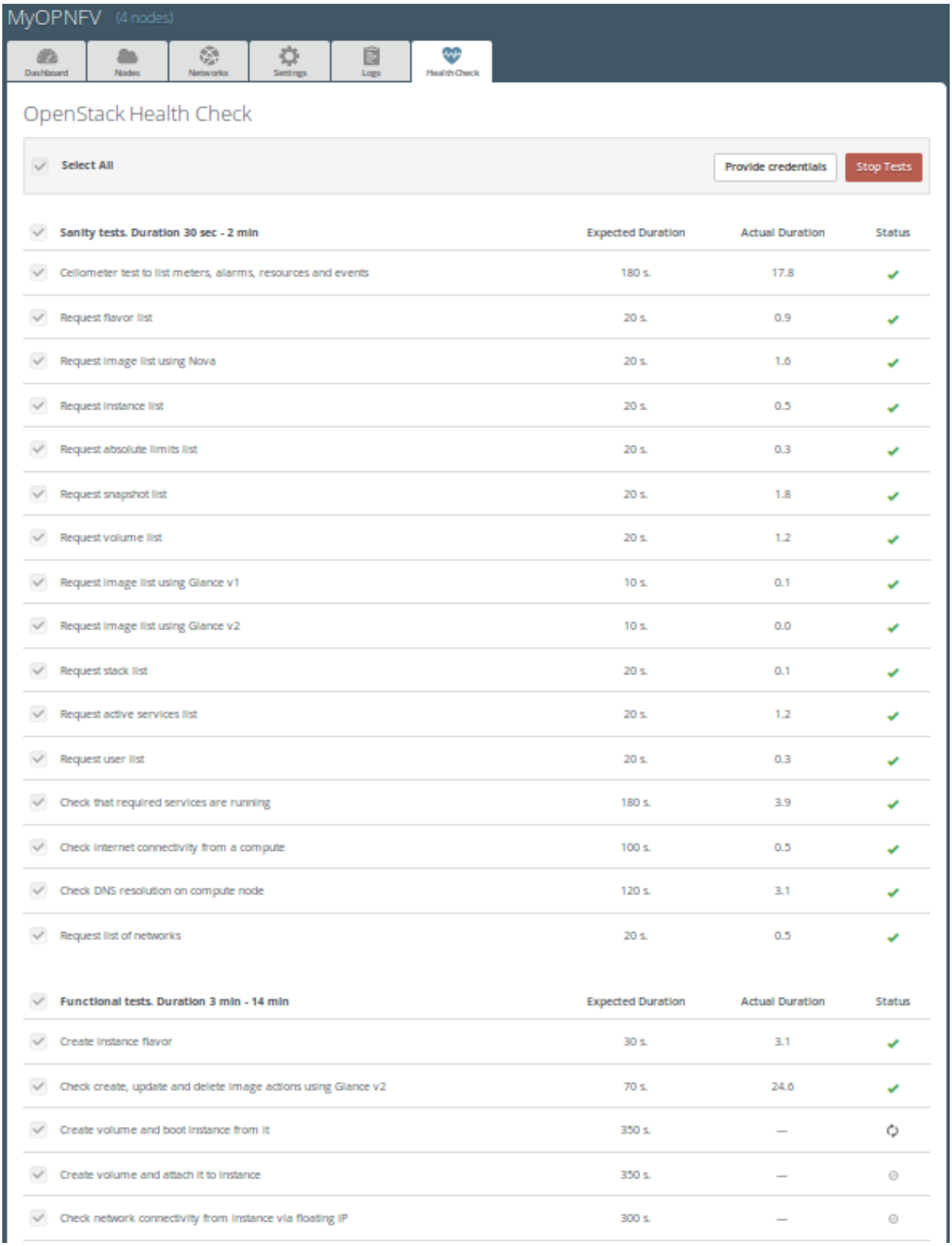
38. Deploy the environment.
    - In the Fuel GUI, click on the "Dashboard" Tab.
    - Click on <Deploy Changes> in the "Ready to Deploy?" section
    - Examine any information notice that pops up and click <Deploy>

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

1.9 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


MyOPNFV (4 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
Network Settings (Neutron with tunneling segmentation)
Add New Node Network Group
Node Network Groups
default
Settings
Neutron L2
Neutron L3
Other
Network Verification
Connectivity Check
Connectivity Check
Network verification checks the following:
1. L2 connectivity checks between nodes in the environment.
2. DHCP discover check on all nodes.
3. Repository connectivity check from the Fuel Master node.
4. Repository connectivity check from the Fuel Slave nodes through the public & admin (PXE) networks.
Verify Networks
Verification succeeded. Your network is configured correctly.
Cancel Changes
Save Settings


MyOPNFV (4 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
OpenStack Health Check
Select All
Provide credentials
Stop Tests
Sanity tests. Duration 30 sec - 2 min
Expected Duration
Actual Duration
Status
Cellometer test to list meters, alarms, resources and events
180 s.
17.8
Request flavor list
20 s.
0.9
Request image list using Nova
20 s.
1.6
Request instance list
20 s.
0.5
Request absolute limits list
20 s.
0.3
Request snapshot list
20 s.
1.8
Request volume list
20 s.
1.2
Request image list using Glance v1
10 s.
0.1
Request image list using Glance v2
10 s.
0.0
Request stack list
20 s.
0.1
Request active services list
20 s.
1.2
Request user list
20 s.
0.3
Check that required services are running
180 s.
3.9
Check internet connectivity from a compute
100 s.
0.5
Check DNS resolution on compute node
120 s.
3.1
Request list of networks
20 s.
0.5
Functional tests. Duration 3 min - 14 min
Expected Duration
Actual Duration
Status
Create instance flavor
30 s.
3.1
Check create, update and delete image actions using Glance v2
70 s.
24.6
Create volume and boot instance from it
350 s.
—
Create volume and attach it to instance
350 s.
—
Check network connectivity from instance via floating IP
300 s.
—

1.10 References

1.10.1 OPNFV

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

1.10.2 OpenStack

3. OpenStack Mitaka Release artifacts
4. OpenStack documentation

1.10.3 OpenDaylight

5. OpenDaylight artifacts

1.10.4 Fuel

6. The Fuel OpenStack project
7. Fuel documentation overview
8. Fuel Installation Guide
9. Fuel User Guide
10. Fuel Developer Guide
11. Fuel Plugin Developers Guide
12. Fuel OpenStack Hardware Compatibility List

1.10.5 Fuel in OPNFV

13. OPNFV Installation instruction for the Colorado release of OPNFV when using Fuel as a deployment tool
14. OPNFV Build instruction for the Colorado release of OPNFV when using Fuel as a deployment tool
15. OPNFV Release Note for the Colorado release of OPNFV when using Fuel as a deployment tool