# CONTENTS

1 About This Document .............................................................................................................. 4  
1.1 Validity of this Document ................................................................................................. 4  
1.2 Audience .......................................................................................................................... 4  
1.3 Technical Support .............................................................................................................. 4  
1.4 Specifics of this Document .............................................................................................. 4  
2 Release Notes ....................................................................................................................... 5  
3 LIMITATIONS ....................................................................................................................... 6  
4 INTEGRATION OVERVIEW ............................................................................................... 7  
  4.1 VSP Integration with Mirantis Fuel .................................................................................. 7  
   4.1.1 Terminology ................................................................................................................ 7  
   4.1.2 Nuage Plugin Overview ............................................................................................... 7  
   4.1.3 Requirements ............................................................................................................. 8  
   4.2 Solution Overview ......................................................................................................... 8  
    4.2.1 Virtualized Services Directory (VSD) ...................................................................... 9  
    4.2.2 Virtualized Services Controller (VSC) ..................................................................... 9  
    4.2.3 Virtual Routing and Switching (VRS) ...................................................................... 9  
    4.2.4 Virtual Routing Services Gateway (VRS-G) ............................................................ 9  
    4.2.5 Nuage Networks OpenStack Neutron Plugin (Liberty) ............................................ 9  
    4.2.6 OpenStack Heat Support for Nuage Extensions ....................................................... 9  
    4.2.7 OpenStack Horizon Support for Nuage Extensions .................................................. 9  
5 PLUGIN INSTALLATION ...................................................................................................... 10  
5.1 Prerequisites .................................................................................................................... 10  
5.2 Installing the Nuage Plugin ............................................................................................. 11  
6 FUEL PLUGIN CONFIGURATION ...................................................................................... 13  
6.1 Configuring the Nuage Plugin .......................................................................................... 13  
7 USING MIRANTIS FUEL WITH VSP AND OPENSTACK ............................................. 22  
7.1 Nuage VSP and Nuage Fuel Plugin Integration Testing .................................................. 22  
8 TROUBLESHOOTING ........................................................................................................ 27
1 About This Document

For a complete list of applicable user documentation, see the Technical Publications section of the Release Notes for your Nuage Networks software version.

1.1 Validity of this Document

Printed versions of this document may not be up to date. Only the Web version of this document is current.

1.2 Audience

This manual is intended for enterprise system administrators who are responsible for enterprise network configuration and administrators for the Nuage VSP/VNS software.

1.3 Technical Support

If you purchased a service agreement for your Nuage Networks VSP/VNS solution and related products from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance. If you purchased an Alcatel-Lucent service agreement, contact your welcome center:

- https://www.alcatel-lucent.com/support

OLCS provides registered customers with access to technical support, software downloads, training, documentation, literature, and other related assets for our products and solutions. For assistance with OLCS, including inability to access, contact us as follows:

- Outside the U.S.: 1-630-224-9000
- Via email: olcshelp@alcatel-lucent.com

1.4 Specifics of this Document

This document provides instructions for installing, configuring and using Nuage Networks plugin for Fuel.

Note:

Mirantis and Mirantis Fuel are trademarks of Mirantis Corporation.
2 Release Notes

1. The User Interface for Nuage Network VSP plugin for Fuel 8.0 has been modified and some of
   the parameters required for deployment have been made mandatory. In order to avoid
   deployment failures due to missing parameters, Nuage Networks specific parameters such as
   CMS ID, VSD IP address, VSC IP address have been marked as mandatory fields in UI for plugin
   configuration.

2. Nuage Network VSP plugin for Fuel 8.0 has to deployed with Nuage VSP 4.0 packages as
   opposed to Nuage VSP 3.2 packages for Fuel 7.0.

3. Nuage Network VSP plugin for Fuel 8.0 requires certain changes to be made in Fuel Code so as
   to support all the test scenarios. These additions have to made keeping in mind that user may
   want to add controller or compute node to an existing deployment.
3 LIMITATIONS

- Nuage Networks VSP includes a distributed routing and switching component called the VRS (Virtual Router & Switch), which is based on Open vSwitch (OVS) and supports tunneling (VXLAN and GRE) to setup the overlay network. This component replaces the standard OVS on the OpenStack compute node; therefore, you must choose the “VLAN with networking segmentation” option in the Networking Setup portion of the plugin configuration. Nuage Networks VSP supports VXLAN and GRE tunnels, but they are managed by the Nuage Networks VRS instead of the standard OpenStack Compute OVS.
4 INTEGRATION OVERVIEW

4.1 VSP Integration with Mirantis Fuel

4.1.1 Terminology

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSP</td>
<td>Nuage Virtualized Services Platform which forms Nuage's Software Defined Networking (SDN) solution</td>
</tr>
<tr>
<td>VSD</td>
<td>Virtualized Services Directory: Policy and analytics engine of Nuage VSP</td>
</tr>
<tr>
<td>VSC</td>
<td>Virtualized Services Controller: Control plane of Nuage VSP</td>
</tr>
<tr>
<td>VRS</td>
<td>Virtualized Routing and Switching: Forwarding plane of Nuage VSP</td>
</tr>
<tr>
<td>VRS-G</td>
<td>Virtualized Routing Services Gateway</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>CMS</td>
<td>Cloud Management System</td>
</tr>
<tr>
<td>PAT</td>
<td>Port Address Translation</td>
</tr>
<tr>
<td>VXLAN</td>
<td>Virtual Extensible LAN: encapsulation protocol for running an overlay network on existing Layer 3 infrastructure</td>
</tr>
<tr>
<td>vPort</td>
<td>Virtual Port</td>
</tr>
</tbody>
</table>

4.1.2 Nuage Plugin Overview

Nuage plugin for Fuel provides automated deployment of Mirantis OpenStack with the Nuage Networks Virtual Services Platform (VSP) that can be used by Mirantis OpenStack for implementing an OpenStack networking service. The Nuage Networks Virtualized Services Platform (VSP) provides Software Defined Networking capabilities for clouds of all sizes – from small private clouds to the largest public clouds in the world and makes the network as readily consumable as compute resources. Nuage VSP is implemented as a non-disruptive overlay for all existing virtualized and non-virtualized server and network resources. No proprietary or purpose-built hardware is required since all components install in Docker containers, hypervisors or virtual machines.

Features:

- Highly scalable and efficient OpenStack networking service.
- Non-disruptive overlay for all existing virtualized and non-virtualized server and network resources.
- Public network access to virtual machine instances via PAT to Underlay feature and floating IP provisioning.
- Metadata agent proxy for handling metadata requests of clients in Openstack instance.
- Increased data throughput by load balancing and fault-tolerance through NIC aggregation.

<table>
<thead>
<tr>
<th>Component</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuage VSP</td>
<td>Commercial</td>
</tr>
<tr>
<td>Fuel Plugin Nuage</td>
<td>Apache 2.0</td>
</tr>
</tbody>
</table>
4.1.3 Requirements

The plugin has the following requirements for software and hardware:

Table 3.2: Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Version/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel</td>
<td>8.0</td>
</tr>
<tr>
<td>Nuage VSP</td>
<td>4.0 or above; with pre-installed VSD and VSC</td>
</tr>
<tr>
<td>OpenStack</td>
<td>Liberty. Production environments need to have at least 3 controllers as part of their OpenStack cluster in HA mode.</td>
</tr>
</tbody>
</table>

4.2 Solution Overview

Nuage Networks Virtualized Services Platform (VSP) is a Software-Defined Networking (SDN) solution that provides data center (DC) network virtualization and automatically establishes connectivity between compute resources upon their creation. Leveraging programmable business logic and a powerful policy engine, the Nuage VSP provides an open and highly responsive solution that scales to meet the stringent needs of massive multi-tenant DCs. The Nuage VSP is a software solution that can be deployed over an existing DC IP network fabric.

The main components in the Nuage VSP solution are Virtualized Services Directory (VSD) and Virtualized Services Directory Architect (VSD-A), Virtualized Services Controller (VSC), Virtual Routing and Switching (VRS) and Virtual Routing and Switching Gateway (VRS-G). For OpenStack deployments, the solution has an OpenStack Neutron-plugin that is included in the software download bundle provided by Nuage Networks (Reference [link]).
4.2.1 Virtualized Services Directory (VSD)

The Nuage VSD is a programmable policy and analytics engine. It provides a flexible and hierarchical network policy framework that enables IT administrators to define and enforce resource policies in a user-friendly manner. The VSD contains a multi-tenant service directory which supports role-based administration of users, compute, and network resources. It also manages network resource assignments such as IP addresses and ACLs. For the purpose of service assurance, the VSD allows the definition of sophisticated statistics rules such as collection frequencies, rolling averages and samples, as well as Threshold Crossing Alerts (TCA). When a TCA occurs it will trigger an event that can be exported to external systems through a generic messaging bus. Statistics are aggregated over hours, days and months and stored in a Hadoop® analytics cluster to facilitate data mining and performance reporting. The VSD runs as a number of processes in a virtual machine (VM) environment.

4.2.2 Virtualized Services Controller (VSC)

The Nuage VSC is the industry’s most powerful SDN controller. It functions as the robust network control plane for DCs, maintaining a full view of per-tenant network and service topologies. Through the VSC, virtual routing and switching constructs are established to program the network forwarding plane, the Nuage VRS, using the OpenFlow™ protocol. The VSC communicates with the VSD policy engine using Extensible Messaging and Presence Protocol (XMPP). An ejabberd XMPP server/cluster is used to distribute messages between the VSD and VSC entities. Multiple VSC instances can be federated within and across DCs by leveraging MP-BGP.

4.2.3 Virtual Routing and Switching (VRS)

The Nuage VRS component is an enhanced Open vSwitch (OVS) implementation that constitutes the network forwarding plane. It encapsulates and de-encapsulates user traffic, enforcing L2-L4 traffic policies as defined by the VSD. The VRS tracks VM creation, migration and deletion events in order to dynamically adjust network connectivity.

4.2.4 Virtual Routing Services Gateway (VRS-G)

The software-based VRS-G can be run in a dedicated bare-metal server or as a virtual machine, allowing the incorporation of bare-metal assets as virtualized extensions of the data center, as well as providing an exit gateway for the DVRS’ overlay, floating-IP and shared-services traffic.

4.2.5 Nuage Networks OpenStack Neutron Plugin (Liberty)

The Nuage Neutron Plugin allows OpenStack to take advantage of this scale and flexibility. Unlike many other OpenStack Networking plugins, the Nuage Neutron Plugin provides fully distributed L2 and L3 networking, including L2 and L3 network isolation, without requiring centralized routing instances such as the Neutron L3 Agent. The Nuage Neutron Plugin also allows connectivity between OpenStack defined networks and other cloud networks, permitting users to deploy flexible network configurations, including routers and subnets, which are shared between OpenStack and other Cloud Management Systems. This allows Cloud administrators unparalleled flexibility in deploying cloud applications and migrating workloads and applications from one CMS to another.

4.2.6 OpenStack Heat Support for Nuage Extensions

Nuage VSP version 4.0 and higher support OpenStack Heat (Liberty) with Neutron supported APIs, and Nuage extensions including: VSD-managed subnet, Gateway and Application Designer.

4.2.7 OpenStack Horizon Support for Nuage Extensions

Nuage VSP version 4.0 provides Nuage extensions for OpenStack Horizon (Liberty). Supported extensions include: Net Partition, VSD-Managed Subnet, Gateway and Application Designer.
5 PLUGIN INSTALLATION

This section includes the following topics:

- Prerequisites
- Installing the Nuage Plugin

5.1 Prerequisites

Please follow the Mirantis user guide (https://docs.mirantis.com/openstack/fuel/fuel-8.0/) to create an OpenStack cluster using Fuel and the Nuage Networks VSP 4.0 Installation Guide to setup the Nuage Networks VSP. This guide assumes that you have already installed Fuel and that all of the OpenStack and Nuage Networks VSP nodes are discovered and functional, including the Nuage Networks VSD (standalone or cluster), VSC(s) and VRS-G.

It is important to note that for this guide and to install the Nuage Networks Plugin you will need the Nuage Networks VRS and neutron packages for your operating system (rpm/deb). Please contact your Nuage Networks account team to obtain the URL to download the Nuage Networks VSP Software Packages and Installation guides. Here is a list of all the files needed prior to the installation of the Nuage Networks Plugin:

- nuage-openstack-heat_*.*
- nuage-openstack-horizon_*.*
- nuage-openstack-neutron_*.*
- nuage-openstack-neutronclient_*.*
- nuagenetlib_*.*
- nuage-metadata-agent_*.*
- nuage-openstack-upgrade-*.tar.gz (needed for step 5.2.8)

On the Nuage Networks VSD, you must create an admin user for OpenStack, you can use the default username/password in the Installation guide or create a new one “osadmin” for example. However, before proceeding to the next section, you must add that user to the following groups: CMS Group, Operator Group and Root Group. See the Nuage Networks VSP Install Guide for steps on how to assign a user to a group.
### 5.2 Installing the Nuage Plugin

1. Make sure the Fuel Master node is installed and running. Please refer to the official Mirantis user guide for details ([https://docs.mirantis.com/openstack/fuel/fuel-8.0/fuel-user-guide.html](https://docs.mirantis.com/openstack/fuel/fuel-8.0/fuel-user-guide.html)). Then, download the Nuage plugin from the Fuel Plugins Catalog that can be found in the following location: [https://www.mirantis.com/products/openstack-drivers-and-plugins/fuel-plugins/](https://www.mirantis.com/products/openstack-drivers-and-plugins/fuel-plugins/)

   Please refer to the Nuage Networks VSP Install Guide for steps on how to assign a user to a group.

2. Log in to the Fuel node:

   Replace all instances of ‘openvswitch-switch’ by ‘nuage-openvswitch-switch’ in the following files on Fuel Node (only for instances of debian packages).

   `/etc/puppet/modules/l23network/manifests/params.pp`

   `/etc/puppet/modules/neutron/manifests/params.pp`


   ```
   Add the bold line as shown below:

   ```

   ```powershell
evac { 'wait-for-int-br':
       command => "ovs-vsctl br-exists ${neutron_integration_bridge}",
       path => ['/sbin', '/bin', '/usr/bin', '/usr/sbin'],
       try_sleep => 6,
       tries => 10,
       returns => [0, 2, 14],
     }
   ```

   **NOTE:** We need to make these Fuel parameter changes mentioned in above 2 steps only once before the first Nuage plugin enabled deployment.

4. Copy the rpm downloaded at previous step to the Fuel Master node and install the plugin:

   ```powershell
   scp nuage-openstack-fuel-plugin-4.0-4.0.4000-1.noarch.rpm <Fuel MasterNode IP>:/tmp/
   ```

5. Log into the Fuel Master node and install the plugin:

   ```powershell
   ssh <Fuel Master Node IP> fuel plugins --install /tmp/ nuage-openstack-fuel-plugin-4.0-4.0.4000-1.noarch.rpm
   ```

   You should get the following output:

   ```powershell
   Plugin -openstack-fuel-plugin-4.0-4.0.4000-1.noarch.rpm was successfully installed
   ```

6. In order to list the Nuage Fuel plugin installed on the Fuel Master node:

   ```powershell
   fuel plugins --list
   [root@fuel tmp]# fuel plugins --list
   id | name                          | version      | package_version
   --- |------------------------------- |--------------|----------------
   1  | nuage-openstack-fuel-plugin   | 4.0.4000     | 4.0.0
   ```
7. Copy Nuage neutron, VRS debian and Metadata Agent packages (obtained from Nuage Networks by subscription, see Prerequisites above) to the Fuel Master node at /var/www/nailgun/plugins/nuage-openstack-fuel-plugin-4.0/repositories/ubuntu location.

**Note:**

All packages placed at the above location on the Fuel Master node must be compressed into Packages.gz file using the following command before starting the deployment:

```
dpkg-scanpackages ./ | gzip -c -> Packages.gz
```

8. Copy the (nuage-openstack-upgrade-*.tar.gz) containing the scripts to generate a unique CMS ID for a given Openstack cluster within a Nuage Networks VSD organization to the Fuel Master node also. Then generate the CMS ID on the Fuel Master node as follows:

```
mkdir -p <dir-upgrade-scripts>
tar -xzvf nuage-openstack-upgrade-*.tar.gz -C <dir-upgrade-scripts>
cd <dir-upgrade-scripts>
cp /var/www/nailgun/plugins/nuage-openstack-fuel-plugin-4.0/deployment_scripts/configure_vsd_cms_id.py /root/<dir-upgrade-scripts>
python configure_vsd_cms_id.py --server <vsd-ip:8443> --serverauth <vsd-username:vsd-password> --organization <vsd-organization> --auth_resource /me --serverssl True --base_uri /nuage/api/v4_0
```

where:

- **vsd-ip**: VSD IP address
- **vsd-username**: VSD UI login username
- **vsd-password**: VSD UI login password
- **vsd-organization**: VSD organization name

This command will give you a CMS ID that is stored in cms_id.txt in dir-upgrade-scripts.
6  FUEL PLUGIN CONFIGURATION

6.1  Configuring the Nuage Plugin

1. Create a new OpenStack environment with Fuel UI wizard:

![Create a new OpenStack environment](image)

**Fig. 5.1: Creating a new OpenStack environment with the Fuel Wizard**
2. Select QEMU-KVM hypervisor type for your environment:

![Fig. 5.2: Select KVM or QEMU hypervisor type for your environment](image1)

3. Select Neutron with VLAN segmentation network model

![Fig. 5.3: Selecting Neutron with VLAN segmentation network model](image2)
4. This deployment skips using any Ceph storage node as a part of the cluster:

![Fig. 5.4: Ceph storage, use LVM](image1)

5. Now your Mirantis Openstack Environment has been created and is ready to use:

![Fig. 5.5: Finish creating the new OpenStack environment](image2)
6. Now assign roles to your OpenStack nodes as follows:

- At least 3 Controllers for High Availability (HA) deployment (16GB memory/64GB HDD at least)
- At least 2 Computes (16GB memory/64GB HDD at least)

![Fig. 5.6: Assigning roles to your OpenStack nodes](image)

7. Make sure the "Networks" tab reflects using "Neutron with VLAN segmentation" as your network:

![Fig. 5.7: Use Neutron with VLAN segmentation as your network](image)
8. Also, you will see the Fuel Nuage plugin in the list of installed plugins in the "Plugins" tab of your Fuel UI:

![Fig. 5.8: Look for the Fuel Nuage plugin in the list of Installed plugins](image)

Fig. 5.8: Look for the Fuel Nuage plugin in the list of Installed plugins
9. Now open Settings tab of the Fuel Web UI and select the “Fuel Nuage Plugin” tab and enable the plugin by selecting the checkbox under the ‘Other’ tab:

![OpenStack Settings](image.png)

**Fig. 5.9: Select the Fuel Nuage Plugin tab and enable the plugin**
10. Enter the Nuage VSD authentication details as per your VSP environment for the relevant fields shown in the previous screen:

- **Nuage Net Partition name**: default_net_partition_name; parameter required in Nuage plugin file on the OpenStack controller node
- **VSD IP address**: Management IP address of the VSD node in the cluster
- **VSD username**: VSD UI login username (mandatory field, and is the admin user mentioned in the section Prequisites)
- **VSD password**: VSD UI login password (mandatory field, and is for the admin user mentioned in the section Prerequisites)
- **VSD organization name**: VSD Organization name
- **Nuage base uri version**: v4_0 (Since we are using Nuage VSP 4.0)
- **Active VSC IP address**: IP address of the active VSC in the Nuage VSP cluster
- **Backup VSC IP address**: IP address of the standby (backup) VSC in the Nuage VSP cluster

11. To enable Nuage metadata agent services on the OpenStack compute nodes as a part of the Nuage Fuel plugin deployment; the following UI fields must be set:

![Image of UI fields](image-url)

**Fig. 5.10: Enabling the Nuage Metadata Agent**

- **Nuage Metadata agent port number**: Port on which metadata agent listens to metadata requests from the tenant VMs on compute nodes (defaults to 9697 as seen above).

- **Nova metadata port number**: Port on which metadata agent listens to metadata requests coming from compute nodes on the controller nodes (defaults to 8775 as seen above).

- **Nova Region Name**: Region name required to setup metadata agent service (Defaults to “RegionOne”).

- **Nova API Endpoint type**: Defaults to “publicURL” as seen above.
12. On the Fuel UI, populate the CMS ID generated on the Fuel node in section 5.2, step 8 (mandatory value):

![Fig. 5.11: Populating the CMS ID](image)

13. The plugin also provides support for PAT to underlay feature wherein tenant VMs can reach public network without configuring floating IPs. In order to enable this feature; select the PAT to Underlay checkbox on the Fuel UI as shown below. Also provide the interface that needs to be used on the compute nodes as an uplink interface after enabling PAT to Underlay feature. The default value for the uplink interface to be used will be set by the plugin to “br-mgmt” as seen below:

![Fig. 5.12: Enabling PAT to Underlay](image)

14. After we set all the above Fuel UI parameters; we run a Network Verification test under the Networks tab (recommended by Fuel) before we begin the final deployment. If the Network Verification goes fine; we are good to start the deployment:

![Fig. 5.14: Verify the network](image)
If the Nuage VSP and Mirantis Openstack Integration via the Nuage Fuel plugin was successful, we will see a deployment success message on the Fuel UI:

Fig. 5.15: Deployment success message

Note:

Due to architecture and plugin implementation decisions, deploying a Nuage Networks VSP as the SDN solution alongside Mirantis OpenStack means that some health checks are expected to fail:

- Floating IP test – The Nuage Networks VSP leverages the Nuage Networks VRS-G for that functionality, which is not included in the standard HC test configuration and checks.
- Cinder test – The Nuage Networks solution does not require/use a cinder node in our deployment, and therefore, this test is expected to fail.
7 USING MIRANTIS FUEL WITH VSP AND OPENSTACK

This section provides instructions for using Mirantis Fuel with Nuage VSP and OpenStack. This section includes the following topics:

7.1 Nuage VSP and Nuage Fuel Plugin Integration Testing

As a part of testing the integration of Nuage VSP and Mirantis OpenStack; we create multiple tenant networks and subnets. 1 VM is spawned on each of the hypervisor node and attached to a private subnet. After VMs are successfully spawned across the 2 compute nodes as shown in the topology; we verify that each of the tenant VM gets assigned an IP address as seen below:

Here we verify tenant VM 1 and tenant VM 2 get assigned an IP address from their respective private subnets 192.168.10.0/24 & 192.168.20.0/24.

Fig. 6.1: Verifying tenant VM

Fig. 6.2: Verifying tenant VM2
Now we verify that both tenant VMs can ping each other as seen below:

```
$ ping 192.168.10.50
PING 192.168.10.50 (192.168.10.50): 56 data bytes
64 bytes from 192.168.10.50: seq=0 ttl=62 time=9.909 ms
64 bytes from 192.168.10.50: seq=1 ttl=62 time=1.553 ms
64 bytes from 192.168.10.50: seq=2 ttl=62 time=2.525 ms
64 bytes from 192.168.10.50: seq=3 ttl=62 time=1.220 ms
64 bytes from 192.168.10.50: seq=4 ttl=62 time=0.905 ms
64 bytes from 192.168.10.50: seq=5 ttl=62 time=1.031 ms
64 bytes from 192.168.10.50: seq=6 ttl=62 time=1.517 ms
64 bytes from 192.168.10.50: seq=7 ttl=62 time=0.788 ms
64 bytes from 192.168.10.50: seq=8 ttl=62 time=0.788 ms
64 bytes from 192.168.10.50: seq=9 ttl=62 time=3.611 ms
--- 192.168.10.50 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.788/2.312/9.909 ms
```

Fig. 6.3: Verify ping tenant VM1

```
$ ping 192.168.20.50
PING 192.168.20.50 (192.168.20.50): 56 data bytes
64 bytes from 192.168.20.50: seq=0 ttl=62 time=1.585 ms
64 bytes from 192.168.20.50: seq=1 ttl=62 time=1.023 ms
64 bytes from 192.168.20.50: seq=2 ttl=62 time=1.060 ms
64 bytes from 192.168.20.50: seq=3 ttl=62 time=0.875 ms
64 bytes from 192.168.20.50: seq=4 ttl=62 time=19.425 ms
64 bytes from 192.168.20.50: seq=5 ttl=62 time=2.693 ms
64 bytes from 192.168.20.50: seq=6 ttl=62 time=5.745 ms
64 bytes from 192.168.20.50: seq=7 ttl=62 time=6.444 ms
64 bytes from 192.168.20.50: seq=8 ttl=62 time=1.028 ms
64 bytes from 192.168.20.50: seq=9 ttl=62 time=3.522 ms
--- 192.168.20.50 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.785/4.340/19.425 ms
```

Fig. 6.4: Verify ping tenant VM2

We also verify that metadata requests from each of the above tenant VMs get a response from the metadata server as seen below. This verifies the Nuage metadata agent service setup on the compute nodes:
All tenant networks and VMs that were created as a part of the Nuage Fuel Plugin Integration testing can also be seen on the VSD dashboard as pasted below:

If the user enabled PAT to underlay on the Fuel UI prior to the deployment; the setup can be validated for the PAT to underlay feature wherein tenant VMs can reach the public network without VRS-G and floating IP configuration for tenant VMs.

As a part of testing the PAT to Underlay feature; user needs to create domains, networks and subnets on the Nuage VSD using the neutron API calls.
In order to enable PAT to Underlay feature for the configured domain on the VSD:

1. Create an external network:

   ```
   neutron net-create <external-network name> --router:external
   ```

2. Create external subnet:

   ```
   neutron subnet-create --name <external-subnet name> <external-network name> <external-subnet/subnet-mask>
   ```

3. Set external network's ID as the gateway for the configured domain on the VSD:

   ```
   neutron router-gateway-set <VSD domain name> <external network ID from step 1 above>
   ```

The above configuration steps will enable PAT settings on the Nuage VSD for the configured domains and tenant VMs should be able to reach the external world as seen below:

```
$ ipaddr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
     valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether fa:16:3e:29:36:98 brd ff:ff:ff:ff:ff:ff
   inet 192.168.10.254/24 brd 192.168.10.255 scope global eth0
     valid_lft forever preferred_lft forever
$ ping 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=51 time=16.295 ms
64 bytes from 8.8.8.8: seq=1 ttl=51 time=16.393 ms
64 bytes from 8.8.8.8: seq=2 ttl=51 time=24.551 ms
64 bytes from 8.8.8.8: seq=3 ttl=51 time=21.332 ms
64 bytes from 8.8.8.8: seq=4 ttl=51 time=17.230 ms
64 bytes from 8.8.8.8: seq=5 ttl=51 time=16.180 ms
64 bytes from 8.8.8.8: seq=6 ttl=51 time=16.329 ms
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 16.180/16.503/24.551 ms
```

Fig. 6.7: Verifying VM1 connectivity
Fig. 6.8: Verifying VM2 connectivity

```
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether fa:16:3e:9b:1a:8f brd ff:ff:ff:ff:ff:ff
  inet 192.168.29.50/24 brd 192.168.29.255 scope global eth0
    inet6 fe80::f016:3eff:fe9b:1a8f/64 scope link
      valid_lft forever preferred_lft forever

$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=51 time=17.587 ms
64 bytes from 8.8.8.8: seq=1 ttl=51 time=15.800 ms
64 bytes from 8.8.8.8: seq=2 ttl=51 time=15.824 ms
64 bytes from 8.8.8.8: seq=3 ttl=51 time=15.884 ms
64 bytes from 8.8.8.8: seq=4 ttl=51 time=15.964 ms
64 bytes from 8.8.8.8: seq=5 ttl=51 time=16.012 ms

--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 15.824/16.191/17.587 ms
```
8 TROUBLESHOOTING

This section provides instructions on troubleshooting Mirantis Fuel with Nuage VSP and OpenStack.

In the case of issues with the deployment and depending on the specific issue, below are some troubleshooting tips:

1. Nuage Fuel plugin has been implemented using puppet and for puppet issues; refer to astute logs under /var/log/docker-logs/astute/astute.log on the Fuel node.
2. For any issues pertaining to Nuage Neutron service; refer to /var/log/neutron/server.log on the OpenStack controller.
3. For issues related to the Nuage Networks VRS, refer to nova-scheduler.log, nova-conductor.log and nova-compute.log under /var/log/nova on the OpenStack compute nodes to troubleshoot.