



MIRANTIS

Integrate Compute Nodes with Red Hat Family Distributions Solution Guide

version 9.2

Contents

| | |
|---|-----------|
| Copyright notice | 1 |
| Preface | 2 |
| Intended Audience | 2 |
| Documentation History | 2 |
| Introduction | 3 |
| Before you integrate the compute nodes with Red Hat family distributions | 4 |
| Limitations and supported configurations | 5 |
| Network configuration limitations | 7 |
| Examples of network configurations | 8 |
| Validate the compute nodes | 11 |
| Prepare nodes for integration | 13 |
| Configure the astute.yaml file parameters | 18 |
| About network objects in the astute.yaml file | 28 |
| Examples | 28 |
| Deploy the compute nodes with Red Hat family distributions | 30 |
| Validate deployment | 34 |
| Segregate workloads | 41 |
| Example of an environment with segregation | 41 |

Copyright notice

2017 Mirantis, Inc. All rights reserved.

This product is protected by U.S. and international copyright and intellectual property laws. No part of this publication may be reproduced in any written, electronic, recording, or photocopying form without written permission of Mirantis, Inc.

Mirantis, Inc. reserves the right to modify the content of this document at any time without prior notice. Functionality described in the document may not be available at the moment. The document contains the latest information at the time of publication.

Mirantis, Inc. and the Mirantis Logo are trademarks of Mirantis, Inc. and/or its affiliates in the United States and other countries. Third party trademarks, service marks, and names mentioned in this document are the properties of their respective owners.

Preface

This documentation provides information on how to use Fuel to deploy OpenStack environments. The information is for reference purposes and is subject to change.

Intended Audience

This documentation is intended for OpenStack administrators and developers; it assumes that you have experience with network and cloud concepts.

Documentation History

The following table lists the released revisions of this documentation:

| Revision Date | Description |
|------------------|-------------|
| February 6, 2017 | 9.2 GA |

Introduction

If your organization owns servers based on Red Hat family distributions, you may want to integrate the servers as compute nodes into your OpenStack environment. This section describes how to prepare the servers and an OpenStack environment for the integration, as well as discusses the Fuel configuration limitations.

This solution guide includes the following topics:

- [Before you integrate the compute nodes with Red Hat family distributions](#)
- [Limitations and supported configurations](#)
- [Network configuration limitations](#)
- [Examples of network configurations](#)
- [Validate the compute nodes](#)
- [Prepare nodes for integration](#)
- [Configure the astute.yaml file parameters](#)
- [About network objects in the astute.yaml file](#)
- [Deploy the compute nodes with Red Hat family distributions](#)
- [Validate deployment](#)
- [Segregate workloads](#)

Before you integrate the compute nodes with Red Hat family distributions

Before you integrate the compute nodes with Red Hat family Linux distributions into an OpenStack environment, verify that you have completed the following:

1. Installed and configured the Fuel Master node as described in the Fuel Installation Guide.
2. Verified that the OpenStack environment that you deployed or plan to deploy meets the configuration requirements described in [Limitations and supported configurations](#) and in [Network configuration limitations](#).
3. Deployed an OpenStack environment as described in the Fuel User Guide.
4. Installed and configured a supported version of Red Hat family Linux distributions on the servers that you plan to use as compute nodes.

Seealso

- [Limitations and supported configurations](#)

Limitations and supported configurations

Fuel supports most configurations for the compute nodes with Red Hat family distributions, except for a few limitations. These compute nodes cannot combine roles, therefore, you can only assign the Compute role to them.

Supported versions of Red Hat family distributions

| Operating System | Version |
|--------------------------|---------|
| Red Hat Enterprise Linux | 7.x |
| Oracle Linux | 7.x |

Supported network configurations

| Topology | Supported | Comment |
|-------------------------------------|-----------|--|
| Neutron with VLAN segmentation | Yes | The Mellanox drivers, SR-IOV, DPDK, Nova CPU Pinning, Nova HugePages are not supported. DVR, L3 HA, L2 Population are supported. |
| Neutron with tunneling segmentation | Yes | The Mellanox drivers, SR-IOV, DPDK, Nova CPU Pinning, Nova HugePages are not supported. DVR, L3 HA, L2 Population are supported. |
| Nova-network | No | VMware vCenter environments are not supported. |

Supported storage configurations

| Storage component | Supported | Comment |
|-------------------|-----------|----------------------|
| Cinder | Yes | |
| Ceph | Yes | |
| Swift | Yes | |
| Radosgw | Yes | Ceph must be enabled |

Supported node roles on Ubuntu nodes

| Role | Supported | Comment |
|-------------------------------|-----------|---------|
| Controller | Yes | |
| Telemetry - MongoDB | Yes | |
| Storage - Cinder | Yes | |
| Storage - Cinder Block Device | Yes | |

| | | |
|--------------------|-----|--|
| Storage - Ceph-OSD | Yes | |
| Ironic | Yes | |
| Compute | Yes | |

Network configuration limitations

If you plan to integrate your existing nodes with Red Hat family distributions as compute nodes to your new OpenStack environment, consider the following network limitations for all networks that Fuel deploys:

- Avoid using the entire range of a Classless Inter-Domain Routing (CIDR) block as IP pool for Public network that provides IPs for Fuel nodes. The pool of IP addresses that you use for Fuel nodes must not interfere with the IP addresses of the compute nodes with Red Hat family distributions. However, the IP addresses for the Fuel nodes and IP addresses for non-Ubuntu compute nodes must be from the same CIDR.
- Limit the range of IP addresses for Public, Management, and Storage networks to 20 IP addresses.

Example:

- Public network: 172.16.0.2 — 172.16.0.21
- Storage network: 192.168.1.2 — 192.168.1.21
- Management network: 192.168.0.2 — 192.168.0.21

To avoid IP address overlapping between the Fuel nodes and the compute nodes with Red Hat family distributions, allocate all other IP addresses to the compute nodes.

Note

Fuel does not control the IP addresses that you assign for the compute nodes and does not identify the overlapping IP addresses. Therefore, you must verify that you have assigned unique IP addresses to the non-Ubuntu compute nodes.

- Assign separate network interfaces for the Fuel Admin and Private networks if you use VLAN segmentation and cannot connect the compute nodes to the Fuel Admin network for security or any other reasons.

For example, if you use the default eth0 interface for Admin network, assign one of other available interfaces, such as eth1 or eth2, for Private network.

Examples of network configurations

This section includes examples of network configuration that you may use for the compute nodes with Red Hat family distributions. You can adjust the provided scenarios as needed.

Example 1

| Network parameter | Description |
|--|--|
| Fuel network configuration includes: | <ul style="list-style-type: none"> • Neutron with tunneling segmentation • Neutron L2 population enabled • Neutron DVR enabled • VLAN tagging enabled |
| Node network groups configuration: | <ul style="list-style-type: none"> • Public network: <ul style="list-style-type: none"> • CIDR: 172.16.0.0/24 • IP Range: 172.16.0.2 — 172.16.0.21 • Gateway: 172.16.0.1 • Use VLAN tagging: No • Storage network: <ul style="list-style-type: none"> • CIDR: 192.168.1.0/24 • IP range: 192.168.1.2 — 192.168.1.21 • Use VLAN tagging: Yes, 102 • Management network: <ul style="list-style-type: none"> • CIDR: 192.168.0.0/24 • IP Range: 192.168.0.2 — 192.168.0.21 • Use VLAN tagging: Yes, 101 |
| Neutron L2 configuration | <ul style="list-style-type: none"> • Tunnel ID range: Start: 2 — End: 65535 |
| Storage configuration | <ul style="list-style-type: none"> • Ceph RBD for volumes (Cinder) |
| Examples of the configuration files with tunneling segmentation and Neutron DVR: | Without connection to the Fuel Admin network astute-tun-dvr.yaml With connection to the Fuel Admin network astute-tun-dvr-admin.yaml |

Example 2

| Network parameter | Description |
|-------------------|-------------|
|-------------------|-------------|

| | |
|--|--|
| Fuel network configuration includes: | <ul style="list-style-type: none"> • Neutron with tunneling segmentation |
| Node network groups configuration: | <ul style="list-style-type: none"> • Public network: <ul style="list-style-type: none"> • CIDR: 172.16.0.0/24 • IP Range: 172.16.0.2 — 172.16.0.21 • Gateway: 172.16.0.1 • Use VLAN tagging: No • Storage network: <ul style="list-style-type: none"> • CIDR: 192.168.1.0/24 • IP range: 192.168.1.1 — 192.168.1.21 • Use VLAN tagging: No • Management network: <ul style="list-style-type: none"> • CIDR: 192.168.0.0/24 • IP Range: 192.168.0.1 — 192.168.0.21 • Use VLAN tagging: No |
| Neutron L2 configuration | <ul style="list-style-type: none"> • Tunnel ID range: Start: 2 — End: 65535 |
| Storage configuration | <ul style="list-style-type: none"> • Ceph RBD for volumes (Cinder) |
| Example of the configuration file with tunneling segmentation: | Without connection to the Fuel Admin network astute-tun-simple.yaml With connection to the Fuel Admin network astute-tun-simple-admin.yaml |

Example 3

| Network Parameter | Description |
|--------------------------------------|--|
| Fuel network configuration includes: | <ul style="list-style-type: none"> • Neutron with VLAN segmentation • VLAN tagging enabled |

| | |
|---|--|
| <p>Node network groups configuration:</p> | <ul style="list-style-type: none"> • Public network: <ul style="list-style-type: none"> • CIDR: 172.16.0.0/24 • IP Range: 172.16.0.2 — 172.16.0.21 • Gateway: 172.16.0.1 • Use VLAN tagging: No • Storage network: <ul style="list-style-type: none"> • CIDR: 192.168.1.0/24 • IP range: 192.168.1.1 — 192.168.1.21 • Use VLAN tagging: Yes, 102 • Management network: <ul style="list-style-type: none"> • CIDR: 192.168.0.0/24 • IP Range: 192.168.0.1 — 192.168.0.21 • Use VLAN tagging: Yes, 101 |
| <p>Neutron L2 configuration</p> | <ul style="list-style-type: none"> • VLAN ID range: Start: 1000 - End: 1030 |
| <p>Storage configuration</p> | <ul style="list-style-type: none"> • Cinder LVM over iSCSI for volumes |
| <p>Example of the configuration file with VLAN segmentation and a bond.</p> | <p>Without connection to the Fuel Admin network astute-vlan-bond.yaml With connection to the Fuel Admin network astute-vlan-bond-admin.yaml</p> |

Validate the compute nodes

Before you add the compute nodes with Red Hat family distributions, you must verify that the nodes meet Fuel requirements. Validation includes disk, network, domain names resolution, and repository connectivity verification. If you use the RHEL compute nodes, you must also verify subscription.

Complete validation steps on each compute node that you plan to add.

Fuel requirements for the non-Ubuntu compute nodes

| Requirement | Description |
|---|--|
| Disk partition | <ul style="list-style-type: none"> • If you have separate partitions: <ul style="list-style-type: none"> • Root file system (/) must have at least 10 GB of disk space. • /var/log must have at least 10 GB of disk space. • /var/lib/nova uses the rest of disk space. You must allocate at least 30 GB of free space. • If you have a single partition, assign at least 50 GB. |
| Network | Configuration of networking equipment connected to the compute nodes must match networking configuration in the Fuel environment. Connection to the Fuel Admin (PXE) network is not required. |
| Domain name resolution | The compute nodes must resolve domain names. |
| RHEL subscription | The RHEL compute nodes must have a valid RHEL subscription. If you use other types of compute nodes, skip this step. |
| Access to the Mirantis OpenStack repository | The compute nodes must have access to the Mirantis OpenStack repository over the Internet or to a local repository mirror available in the company's internal network. |

To validate the compute nodes:

1. Verify network by ensuring that all networking equipment, such as VLANs, ports, and so on is properly connected and configured.
2. Log in to a compute node.
3. Verify disk partition by typing:

```
df -h
```

Example of system response:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/os-root  15G  2.2G  12G  16% /
/dev/vda3        196M  44M  143M  24% /boot
/dev/mapper/os-log  10G  100M  10G   1% /var/log
/dev/mapper/vm-nova  50G   33M  50G   1% /var/lib/nova
```

Partitions are specified under the Mounted on column. Verify that the size of each partition is equal or exceeds the values specified in the [Fuel requirements for the non-Ubuntu compute nodes](#) table.

4. Validate domain name resolution by verifying that the nameserver specified in the `/etc/resolv.conf` file is the IP address of the DNS server.

Example:

```
# Generated by NetworkManager
search local
nameserver 8.8.8.8
```

5. If you use the RHEL compute node, verify that the subscription is attached by typing:

```
subscription-manager list
```

Example of system response:

```
+-----+
Installed Product Status
+-----+
Product Name:  Red Hat Enterprise Linux Server
Version:      7.1
Arch:         x86_64
Status:       Subscribed
```

6. Verify that the compute node can access the OpenStack repositories:

```
curl <url-to-repo>
```

7. Repeat step 2 — step 6 on each compute node that you plan to add.

Prepare nodes for integration

Before preparing the nodes with the Red Hat family distributions for integration, verify that you have completed the tasks described in [Validate the compute nodes](#). After that, complete the steps described in this section on each non-Ubuntu compute node.

Prerequisites:

- GNU Wget

To prepare nodes for integration:

1. Log in to a compute node with the Red Hat family distribution as root.
2. Enable SSH access for the root or other user with root privileges using public SSH keys.
 1. Create a folder called `/root/.ssh`.
 2. Add your public SSH key to `/root/.ssh/authorized_keys`.
 3. Verify the configuration by logging in to the compute node as root using SSH.

You should be logged in without a password.

3. Create an OpenStack repository file with the following content:

```
[mos-9.0]
name=mos-9.0
type=rpm-md
baseurl=<url-to-repo>
gpgcheck=1
enabled=1
priority=5
```

Specify the `url-to-repo` in the following format:
`http://[FUEL_MASTER_ADMIN_IP]:8080/mos-centos/x86_64/`.

4. Import the repository key using Wget:

```
wget <url-to-key> -P /etc/pki/rpm-gpg/ rpm --import
/etc/pki/rpm-gpg/<repo-key-name>
```

Example:

```
wget http://infra.mirantis.net/mos-repos/os/RPM-GPG-KEY-mos9.0
-P /etc/pki/rpm-gpg/ rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-mos9.0
```

5. Enable additional mirrors by typing:

- for an RHEL compute node:

```
yum install yum-utils -y
yum-config-manager --enable rhel-7-server-rh-common-rpms
yum-config-manager --enable rhel-7-server-extras-rpms
yum-config-manager --enable rhel-7-server-optional-rpms
```

- for an Oracle Linux compute node:

```
yum install yum-utils -y
yum-config-manager --enable ol7_addons
yum-config-manager --enable ol7_optional_latest
```

6. Verify that the repositories have been enabled by typing:

```
yum repolist all
```

7. Ensure the compute node downloads packages from the correct repository by installing the yum-plugin-priorities package:

```
yum makecache
yum install yum-plugin-priorities -y
```

8. Configure Security-Enhanced Linux:

1. Ensure that the latest selinux-policy is installed:

```
yum update selinux-policy
```

2. Install the openstack-selinux package:

```
yum install openstack-selinux
```

9. Generate public SSH keys for the VM migration feature.

1. Create the nova directory in /var/lib/astute:

```
mkdir -p /var/lib/astute/nova
```

- If the OpenStack environment already contains compute nodes, copy the following SSH keys to this compute node:

```
/var/lib/astute/nova/nova
/var/lib/astute/nova/nova.pub
```

- If this compute node is the first compute node that you add to the OpenStack environment, create new SSH public keys for nova by typing:

```
ssh-keygen -q -f /var/lib/astute/nova/nova -N ''
```


10 If you have deployed Ceph, copy the Ceph SSH keys to the compute node.

1. Create the ceph directory in /var/lib/astute:

```
mkdir -p /var/lib/astute/ceph
```

2. Copy the following SSH keys from the Ceph nodes to this compute node:

```
/var/lib/astute/ceph/ceph  
/var/lib/astute/ceph/ceph.pub
```

11 Install Puppet 3.x and Ruby 2.1.x by typing:

```
yum install puppet ruby -y
```

12 Install the netaddr and openstack Ruby gems:

```
yum install rubygem-openstack.noarch rubygem-netaddr.noarch -y
```

13 Install Fuel Puppet tasks and modules for deployment:

```
yum install fuel-library9.0 -y
```

14 If you deploy an Oracle Linux compute node, ensure that the Linux kernel version 3.10 without UEK is set as default.

1. Check all available kernels by typing:

```
grep ^menuentry /boot/grub2/grub.cfg | cut -d '"' -f2
```

Example of output:

```
Oracle Linux Server (3.10.0-327.el7.x86_64 with Linux) 7.2  
Oracle Linux Server (3.8.13-98.7.1.el7uek.x86_64 with Unbreakable Enterprise Kernel) 7.2  
Oracle Linux Server (0-rescue-061cade29aff418680018ed728ece5a1 with Linux) 7
```

2. Set proper kernel as default by typing:

```
grub2-set-default "Oracle Linux Server (3.10.0-327.el7.x86_64 with Linux) 7.2"
```

15 Verify that the KVM modules are enabled.

Depending on the host CPU, you must enable kvm and kvm_intel or kvm_adm modules. The following text is an example for Intel CPU.

1. Verify the status of the kvm module by typing:

```
lsmod | grep kvm
```

Example of output:

```
kvm_intel      148081 3
kvm            461126 1 kvm_intel
```

In the output above, the kvm module is enabled.

2. If the kvm module is disabled, enable it by typing:

```
modprobe kvm-intel
```

3. Enable persistent module loading for kvm-intel by creating the /etc/modules-load.d/kvm-intel.conf file with the following content:

```
# Load kvm-intel.ko at boot
kvm-intel
```

4. Verify if the nested KVM Kernel virtualization is enabled:

```
cat /sys/module/kvm_intel/parameters/nested
```

This command returns N for disabled and Y for enabled.

5. Remove the intel KVM Kernel module:

```
rmmod kvm-intel
```

6. Enable persistent nested virtualization across reboots:

```
echo 'options kvm-intel nested=y' >> /etc/modprobe.d/dist.conf
```

- 16 Reboot the compute node:

```
reboot
```

- 17 Create the /etc/astute.yaml file and configure it according to your OpenStack environment requirements.

- 18 Repeat Step 1 — Step 17 on each compute node with Red Hat family distribution.

Seealso

- [Working with Kernel modules](#)
- [Configure DevStack with KVM-based Nested Virtualization](#)

- [astute.yaml parameters](#)
- [Network scenario example](#)

Configure the astute.yaml file parameters

The Fuel Puppet manifests use the astute.yaml file to deploy and configure compute nodes. Edit the astute.yaml file to address the requirements of your configuration and place it in the /etc/ directory before applying Puppet manifests.

If you deploy additional components, such as Sahara, Murano, or Ceilometer, you must update the astute.yaml network section with the corresponding network roles and parameters. You can copy this information from the astute.yaml file on the controller node.

The following table describes the parameters and fields of the astute.yaml file. All IDs and IP addresses used in this table are provided for the purpose of example. Modify these parameters for your environment as needed.

Parameters of the astute.yaml file

| Field | Definition |
|--------------|--|
| amqp_hosts | List of sockets (ip_address:port pairs) on which RabbitMQ listens for connections. |
| | Example: <pre>amqp_hosts: "192.168.0.6:5673, 192.168.0.7:5673, 192.168.0.8:5673"</pre> |
| debug | Enables debug logging level for OpenStack services. Allowed values: true or false. |
| | Example: <pre>debug: false</pre> |
| external_dns | List of DNS servers. |
| | Example: <pre>external_dns: dns_list: [8.8.8.8, 8.8.4.4]</pre> |
| external_ntp | List of NTP servers. |
| | Example: <pre>external_ntp: ntp_list: [0.fuel.pool.ntp.org, 1.fuel.pool.ntp.org, 2.fuel.pool.ntp.org]</pre> |
| fqdn | Fully Qualified Domain Name (FQDN) of the compute node. |

| | |
|--------------------------|--|
| | <p>Example:</p> <pre>fqdn: rhel.domain.tld</pre> |
| libvirt_type | Type of hypervisor used on the compute node. This value depends on the Fuel environment configuration. Allowed values: kvm or qemu. |
| | <p>Example:</p> <pre>libvirt_type: kvm</pre> |
| management_network_range | Must match the value set for the OpenStack environment. You can find this value in the astute.yaml file on the controller node. |
| | <p>Example:</p> <pre>management_network_range: 192.168.0.0/24</pre> |
| master_ip | <p>IP address of the Fuel Master node.</p> <ul style="list-style-type: none"> • If the compute node has network connectivity to the Fuel's Admin (PXE) network, set this value to the Fuel master node IP in Fuel Admin (PXE) network. • If the compute node does not have network connectivity to the Fuel Admin (PXE) network, set this value to the Fuel Master node public IP address. |
| | <p>Example:</p> <pre>master_ip: 10.20.0.2</pre> |
| mp | Required field. Must be empty. |
| | <p>Example:</p> <pre>mp: {}</pre> |

| | |
|-------------------------|--|
| <p>network_metadata</p> | <p>network_metadata provides information about each OpenStack node network configuration, including the non-Ubuntu compute nodes:</p> <ul style="list-style-type: none"> • key_node1_name — internal immutable name of the OpenStack node. Must be in format node-<code>{uid}</code>. • node1_fqdn — FQDN of the OpenStack node. For example, <code>rhel-node.domain.tld</code>. • node1_name — hostname of the OpenStack node. For example, if FQDN is <code>rhel-node.domain.tld</code>, <code>node1_name</code> is <code>rhel-1</code>. • node1_network_metadata_roles_description List of IP addresses of the OpenStack environment components. All IP addresses must be set as addresses of the related OpenStack node. For example: <div data-bbox="565 737 1481 1020" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>network_roles: {admin/pxe: 10.20.0.1, aodh/api: 192.168.0.3, ceilometer/api: 192.168.0.3, ceph/public: 192.168.1.1, ceph/radosgw: 172.16.164.203, ceph/replication: 192.168.1.1, cinder/api: 192.168.0.3, cinder/iscsi: 192.168.1.1, ex: 172.16.164.203, fw-admin: 10.20.0.1, glance/api: 192.168.0.3, glance/glare: 192.168.0.3, heat/api: 192.168.0.3, horizon: 192.168.0.3, ironic/api: 192.168.0.3, keystone/api: 192.168.0.3}</pre> </div> • unique_node1_id — Unique identifier of the OpenStack node. This ID must be unique and must not match IDs of any other nodes deployed by Fuel or manually added to the OpenStack environment. • vips_description — List of the OpenStack controller virtual IP addresses. Copy this value from the <code>astute.yaml</code> file on the controller node. |
|-------------------------|--|

Example:

```
network_metadata:
  nodes:
    *key_node1_name*:
      fqdn: *node1_fqdn*
      name: *node1_name*
      network_roles:
        *node1_network_metadata_roles_description*
      node_roles: [ceph-osd, primary-controller]
      uid: *unique_node1_id*
      user_node_name: *node1_name*
    *key_node2_name*:
      fqdn: *node2_fqdn*
      name: *node2_name*
      network_roles:
        *node2_network_metadata_roles_description*
      node_roles: [compute]
      uid: *unique_node2_id*
      user_node_name: *node2_name*
  vips:
    *vips_description*
```

| | |
|-----------------------|--|
| <p>network_scheme</p> | <p>network_scheme</p> <p>The content of this section depends on the OpenStack environment network configuration. For more information, see: Network configuration limitations.</p> <ul style="list-style-type: none"> • interfaces <p>List of physical network interfaces on the compute node. The network interface parameters are managed by ethtool.</p> <p>For more information, see: Network interface offloading settings.</p> <p>You can configure the offloading and vendor-specific parameters, such as interface_bus_info and interface_driver.</p> <ul style="list-style-type: none"> • interface_bus_info <p>The network interface bus information (optional parameter). For example, 0000:00:07.0.</p> • interface_driver <p>Network interface driver (optional parameter). For example: e1000.</p> • br_mesh_ip <p>IP address of the compute node in the Fuel Private network. For example, 192.168.2.25/24.</p> • br_mgmt_ip <p>IP address of the compute node in the Fuel Management network. For example, 192.168.0.25/24.</p> • br_storage_ip <p>IP address of the compute node in the Fuel storage network. For example, 192.168.1.25/24.</p> • roles <ul style="list-style-type: none"> • roles_description <p>List of roles and related endpoints of each OpenStack component in the OpenStack environment.</p> • transformations <ul style="list-style-type: none"> • transformations_description — a sequence of actions to build the required network topology. The transformations are applied by the Puppet I23network module and, therefore, must be compatible with the module. For more information, see: About network objects in the astute.yaml file. |
|-----------------------|--|

| | | |
|---------------------------------------|---|--|
| | <p>Example:</p> <pre> network_scheme: endpoints: br-ex: {IP: none} br-floating: {IP: none} br-mesh: IP: [*br_mesh_ip*] br-mgmt: IP: [*br_mgmt_ip*] br-storage: IP: [*br_storage_ip*] interfaces: eth0: ethtool: offload: {} vendor_specific: {bus_info: *interface_bus_info*, driver: *interface_driver*} eth1: ethtool: offload: {} vendor_specific: {bus_info: *interface_bus_info*, driver: *interface_driver*} provider: Inx roles: *roles_description* transformations: *transformations_description* </pre> | |
| <p>neutron_advanced_configuration</p> | <ul style="list-style-type: none"> • neutron_dvr — defines whether Neutron DVR is enabled or not. Allowed values: true or false. • neutron_l2_pop — defines whether the L2 population mechanism is enabled or not. If you use tunneling, enable this option. Allowed values: true or false. • l3_ha — defines whether Level 3 high availability is enabled or not. If you use Neutron DVR, set this value to false. Allowed values: true or false. | |
| | <p>Example:</p> <pre> neutron_advanced_configuration: neutron_dvr: *dvr* neutron_l2_pop: *l2_pop* neutron_l3_ha: *l3_ha* </pre> | |
| <p>neutron_mellanox</p> | <p>The Neutron Mellanox plugin is not supported. Therefore, in OpenStack environments that include compute nodes with Red Hat family distributions this value should always be set to disabled.</p> | |

| | |
|---------------------------|--|
| | <p>Example:</p> <pre>neutron_mellanox: plugin: disabled</pre> |
| nodes | <ul style="list-style-type: none"> • <code>internal_address</code> — IP address of the compute node in the Fuel Management network. For example, 192.168.0.25. • <code>internal_netmask</code> — netmask for the Fuel Management network. For example, 255.255.255.0. • <code>storage_address</code> — unique storage IP address of the compute node. For example, 192.168.1.25. • <code>storage_netmask</code> — netmask for storage network in the Fuel environment. For example, 255.255.255.0. |
| | <p>Example:</p> <pre>nodes: - {fqdn: *fqdn*, internal_address: *internal_address*, internal_netmask: *internal_netmask*, name: *node_name*, role: compute, storage_address: *storage_address*, storage_netmask: *storage_netmask*, uid: *unique_node_id*, user_node_name: *node_name*}</pre> |
| nova | <ul style="list-style-type: none"> • <code>nova_db_password</code> — the nova database password. Copy this value from the controller node. • <code>nova_user_password</code> — the nova user password. Copy this value from the controller node. |
| | <p>Example:</p> <pre>nova: {db_password: *nova_db_password*, state_path: /var/lib/nova, user_password: *nova_user_password*}</pre> |
| nova_quota | <p>Verify Nova quotas in the Fuel web UI and set the <code>nova_quota</code> parameter according to this setting. Allowed values: true or false.</p> |
| | <p>Example:</p> <pre>nova_quota: false</pre> |
| public_network_assignment | <p>Verify Assign public network to all nodes in the Fuel web UI and set the <code>assign_to_all_nodes: false</code> parameter according to this setting. Available values: true or false.</p> |
| | <p>Example:</p> <pre>public_network_assignment: assign_to_all_nodes: false</pre> |
| public_ssl | <p>Required field, must be empty.</p> |

| | |
|------------------|---|
| | <p>Example:</p> <pre>public_ssl: {}</pre> |
| puppet | Location of Puppet files |
| | <p>Example:</p> <pre>puppet: manifests: /etc/puppet/manifests/ modules: /etc/puppet/modules/</pre> |
| quantum | Enables neutron. This setting must be set to true because only Neutron topologies are supported. |
| | <p>Example:</p> <pre>quantum: true</pre> |
| quantum_settings | <p>quantum_settings content of this section depends on network configuration in the Fuel environment. See example in astute-tun-dvr.yaml. For more information, see: Network configuration limitations.</p> <ul style="list-style-type: none"> • keystone_admin_password — the keystone administrator password. Must match the keystone password you set for the OpenStack environment. Copy this value from the astute.yaml file on the controller node. • tunnel_id_ranges — tunnel ID ranges. Verify this value in the Tunnel ID range in the Fuel web UI. • metadata_proxy_shared_secret — this value must match the identical value set in the astute.yaml file on the controller node. |
| | <p>Example:</p> <pre>quantum_settings: database: {} keystone: {admin_password: *keystone_admin_password*} L3: use_namespaces: true L2: phys_nets: physnet1: {bridge: br-floating, vlan_range: null} base_mac: fa:16:3e:00:00:00 segmentation_type: tun tunnel_id_ranges: *tunnel_id_ranges* metadata: {metadata_proxy_shared_secret: *metadata_proxy_shared_secret*}</pre> |

| | |
|-----------------------|--|
| rabbit | rabbit_password — password for the RabbitMQ user. The value must match the same value set for the OpenStack environment. Copy this value from the astute.yaml file on the controller node. |
| | <p>Example:</p> <pre>rabbit: {password: *rabbit_password*}</pre> |
| roles | Role of the compute node. This value must be always set to [compute]. |
| | <p>Example:</p> <pre>roles: [compute]</pre> |
| storage | Storage configuration must match the storage configuration set for the OpenStack environment. You can copy this configuration from the astute.yaml file on the controller node. |
| | <p>Example:</p> <pre>storage: admin_key: AQAOUQ9XAAAAABAASWpxfuzqTjjhQGTqumDtdQ== bootstrap_osd_key: AQAOUQ9XAAAAABAaHmWisnDSf90zjQ0ntfaMw== ephemeral_ceph: true fsid: 79107512-c5a2-4b1b-9b03-630ccd1fa16d images_ceph: true images_vcenter: false metadata: {group: storage, label: Storage Backends, weight: 60} mon_key: AQAOUQ9XAAAAABAafdUqtH7kdJz10AFpDYbfkQ== objects_ceph: true osd_pool_size: '3' per_pool_pg_nums: { .rgw: 64, backups: 64, compute: 64, default_pg_num: 64, im volumes: 64} pg_num: 64 radosgw_key: AQAOUQ9XAAAAABAAnk4QTG+rMwdJD7+nbv3A== volumes_block_device: false volumes_ceph: true volumes_lvm: false</pre> |
| storage_network_range | Storage network range must match the storage network range set for the OpenStack environment. You can copy this configuration from the astute.yaml file on the controller node. |
| | <p>Example:</p> <pre>storage_network_range: 192.168.1.0/24</pre> |
| uid | Unique identifier of the node. This value must not be different from any other node IDs for nodes deployed by Fuel or manually. |

| | |
|----------------|--|
| | <p>Example:</p> <pre>uid: '25'</pre> |
| use_cow_images | <p>Defines whether to use the qcow images or not. Verify the setting of Use qcow format for images in the Fuel web UI and the values accordingly. Allowed values: true or false.</p> |
| | <p>Example:</p> <pre>use_cow_images: true</pre> |
| use_vcenter | <p>VMware vCenter is not supported. Therefore, this setting should be set to false.</p> |
| | <p>Example:</p> <pre>use_vcenter: false</pre> |
| user_node_name | <p>Name of the compute node. For example, if FQDN of the compute node is rhel.domain.tld, then node_name can be rhel.</p> |
| | <p>Example:</p> <pre>user_node_name: rhel</pre> |

Seealso

- Example of the Astute configuration for an environment with 50 RHEL compute nodes: [astute-vlan-ceph-50-nodes.yaml](#)
- [About network objects in the astute.yaml file](#)

About network objects in the astute.yaml file

To use physical network interfaces in different OpenStack environment networks, such as Storage, Management, and so on, you must create network objects, such as network bonds or ports and associate them with the corresponding network bridges. You can specify this configuration in the transformations section of the astute.yaml file. You can specify multiple network objects in one file.

Parameters in the transformations section depend on the action that the transformation performs. The following table lists some of the parameters that you can add in the transformations section:

Transformation parameters

| Parameter | Description |
|-----------|--|
| action | Required parameter. Determines the action that the transformation performs. Example: add-br, add-port, add-patch, add-bond. |
| name | Name of the new network object. Example: br-mgmt, bond0, eth0. |
| provider | Name of the resource provider, such as Linux bridge or Open vSwitch (OVS) bridge. If you do not specify provider, Linux bridge is used by default. Example: ovs. |

Examples

- Configure a management bridge for a physical network interface with a VLAN.

```
- action: add-br
  name: br-mgmt
```

This action creates a simple bridge without ports with Linux bridge as a provider.

- Configure a management bridge with Open vSwitch as a provider:

```
- action: add-br
  name: br-floating
  provider: ovs
```

- Add a VLAN interface p2p1.101 to the br-mgmt bridge:

```
- action: add-port
  bridge: br-mgmt
  name: p2p1.101
```

- Add a patch between the br-ex Linux bridge and the br-floating OVS bridge with MTU 65000:

```
- action: add-patch
  bridges:
  - br-floating
  - br-ex
  mtu: 65000
```

- Add a bond of eth3 and eth5 for the br-aux bridge with the balance-rr mode.
- Neutron with VLAN segmentation uses an additional OVS bridge called br-prv. You must associate this bridge with a physical network interface through a Linux bridge. If the physical network interface does not contain a Linux bridge, you must add it.

```
- action: add-br
  name: br-aux
- action: add-patch
  bridges:
  - br-prv
  - br-aux
  mtu: 65000
- action: add-port
  bridge: br-aux
  name: eth1
```

Deploy the compute nodes with Red Hat family distributions

Before deploying the compute nodes with Red Hat family distributions, you must complete the tasks described in the [Validate the compute nodes](#) and [Prepare nodes for integration](#) sections.

The process of the compute nodes deployment involves applying Puppet manifests to the nodes configuration in a specific order.

For the purpose of example, all Puppet manifests are located in `/etc/puppet/modules/`.

The following table describes the Puppet manifests that you must apply on each compute node with Red Hat family Linux distribution.

Puppet manifests for the compute nodes

| Puppet manifest | Description | Path to file |
|--------------------------------|--|---|
| <code>hiera.pp</code> | Configures the hiera packages and its dependencies. | <code>/etc/puppet/modules/osnailyfacter/modular/hiera/hiera.pp</code> |
| <code>globals.pp</code> | Optimizes the hiera configuration file structure for Puppet. | <code>/etc/puppet/modules/osnailyfacter/modular/globals/globals.pp</code> |
| <code>firewall.pp</code> | Configures firewall to accept connections from the OpenStack components. | <code>/etc/puppet/modules/osnailyfacter/modular/firewall/firewall.pp</code> |
| <code>tools.pp</code> | Adds the following tools for debugging and deployment: man, atop, tmux, screen, tcpdump, strace. | <code>/etc/puppet/modules/osnailyfacter/modular/tools/tools.pp</code> |
| <code>netconfig.pp</code> | Configures network interfaces and bridges according to the settings specified in the following sections of the <code>astute.yaml</code> file: <ul style="list-style-type: none"> <code>network_metadata</code> (incoming data) <code>network_scheme</code> <code>transformations</code> | <code>/etc/puppet/modules/osnailyfacter/modular/netconfig/netconfig.pp</code> |
| <code>/roles/compute.pp</code> | Installs required packages for nova-compute. Configures the libvirt and nova-compute services. | <code>/etc/puppet/modules/openstack_tasks/examples/roles/compute.pp</code> |

| | | |
|----------------------------|---|---|
| common-config.pp | Installs and configures required packages for Neutron. | /etc/puppet/modules/openstack_tasks/examples/openstack-network/common-config.pp |
| ml2.pp | Configures Neutron ML2 plugin and service. | /etc/puppet/modules/openstack_tasks/examples/openstack-network/plugins/ml2.pp |
| l3.pp | (Optional) If you use Neutron DVR, this manifest configures Neutron L3 agent on the compute node. Otherwise, do not apply this manifest. | /etc/puppet/modules/openstack_tasks/examples/openstack-network/agents/l3.pp |
| metadata.pp | (Optional) If you enabled Neutron DVR, configures the Neutron metadata agent on the compute node. Otherwise, do not apply this manifest. | /etc/puppet/modules/openstack_tasks/examples/openstack-network/agents/metadata.pp |
| compute-nova.pp | Applies common configuration for Nova and Neutron. Starts the nova-compute service. | /etc/puppet/modules/openstack_tasks/examples/openstack-network/compute-nova.pp |
| enable_compute.pp | Configures the nova-compute service to start on boot. | /etc/puppet/modules/openstack_tasks/examples/roles/enable_compute.pp |
| /dns/dns_client.pp | Sets DNS resolver to get DNS information from Openstack controllers. | /etc/puppet/modules/osnailyfacter/modular/dns/dns-client.pp |
| configure_default_route.pp | (Optional) Checks if fw-admin endpoint has gateway and any network includes management vrouter vip address. If yes - it removes it and sets default route to vip of vrouter via management network. | /etc/puppet/modules/osnailyfacter/modular/netconfig/configure_default_route.pp |
| /ntp/ntp-client.pp | Sets ntp to synchronize time from Openstack controllers. | /etc/puppet/modules/osnailyfacter/modular/ntp/ntp-client.pp |
| /ceilometer/compute.pp | (Optional) If you deploy Ceilometer, applies the Ceilometer configuration to the compute node. Otherwise, do not apply this manifest. | /etc/puppet/modules/openstack_tasks/examples/ceilometer/compute.pp |

| | | |
|-----------------------|---|--|
| /ceph/ceph_compute.pp | (Optional) If you deploy Ceph, applies Ceph configuration to the compute node. Otherwise, do not apply this manifest. | /etc/puppet/modules/osnailyfactor/modular/ceph/ceph_compute.pp |
|-----------------------|---|--|

To deploy the compute nodes with Red Hat family distributions:

1. Apply the hiera.pp Puppet manifests by typing:

```
puppet apply -vd --logdest syslog <path-to-manifest.pp>
```

Example:

```
puppet apply -vd --logdest syslog
/etc/puppet/modules/osnailyfactor/modular/hiera/hiera.pp
```

2. Apply the globals.pp Puppet manifest using the command in step 1.

3. Apply the firewall.pp Puppet manifest:

- If the compute node was configured without access to any custom internal networks except for the default internal network and Fuel Admin (PXE) network, apply the firewall.pp Puppet manifest using the command in step 1.
- If the compute node has access to custom Fuel internal networks instead of the Fuel Admin (PXE) network, modify the command in step 1 so it allows access to the corresponding internal networks:

```
puppet apply -vd --logdest syslog
/etc/puppet/modules/osnailyfactor/modular/firewall/firewall.pp \
&& iptables -I INPUT 1 -s <network> -p tcp -m multiport --ports 22 -m
comment --comment "ssh from <network>" -j ACCEPT \
&& service iptables save;
```

Example:

```
puppet apply -vd --logdest syslog
/etc/puppet/modules/osnailyfactor/modular/firewall/firewall.pp \
&& iptables -I INPUT 1 -s 192.168.122.0/24 -p tcp -m multiport --ports
22 -m comment --comment "ssh from 192.168.122.0/24" -j ACCEPT \
&& service iptables save;
```

4. Apply the tools.pp Puppet manifest using the command in step 1.

5. Apply the netconfig.pp Puppet manifest using the command in step 1.

You should now be able to access the controller node through Management and Storage networks.

6. Verify network configuration by pinging the controller node through the Storage and Management networks.

7. Apply the remaining Puppet manifests in the following order using the command in step 1:

- /roles/compute.pp
- common-config.pp
- ml2.pp
- l3.pp
- metadata.pp
- compute-nova.pp
- enable_compute.pp
- /dns/dns_client.pp
- /ntp/ntp-client.pp

8. Log in to any OpenStack node.

9. Add the list of management IP addresses and host names to the /etc/hosts file.

10 Repeat previous step on each OpenStack node:

• Example:

```
127.0.0.1 localhost
192.168.0.3 node-4.domain.tld node-4
192.168.0.3 messaging-node-4.domain.tld messaging-node-4
192.168.0.4 messaging-node-2.domain.tld messaging-node-2
192.168.0.4 node-2.domain.tld node-2
192.168.0.5 node-3.domain.tld node-3
192.168.0.5 messaging-node-3.domain.tld messaging-node-3
192.168.0.25 rhel-1.domain.tld rhel-1
192.168.0.25 messaging-rhel-1.domain.tld messaging-rhel-1
```

11 If you deployed OpenStack with Ceph, apply /ceph/ceph_compute.pp on each compute node with Red Hat family Linux distribution.

12 If you deployed OpenStack with ceilometer, apply /ceilometer/compute.pp on each compute node with Red Hat family Linux distribution.

13 If you deployed a non-Ubuntu compute node with access to the Fuel Admin network and want the default gateway of the node to point to the virtual router of controller cluster, apply configure_default_route.pp.

14 Proceed to [Validate deployment](#).

Validate deployment

After you complete the tasks described in [Deploy the compute nodes with Red Hat family distributions](#), you must verify your deployment.

Your deployment must meet the following requirements:

Validation requirements for compute nodes with Red Hat family distributions

| Validation | Requirement |
|--|--|
| Verify the state of OpenStack services. | <ul style="list-style-type: none"> • The compute node is up and enabled in the nova hypervisor list on the controller node. • The nova-compute service on the compute node is up and enabled in the nova services list. • The status of the OVS agent on the compute node is alive in the list of neutron agents. |
| Verify the launch of a virtual machine instance on the compute node. | <p>After the launch, the output of the nova list command returns the following properties for the created instance:</p> <ul style="list-style-type: none"> • The Status column has the value ACTIVE. • The Power State column has value Running. <p>The output of nova show command returns the following properties set to hostname of the compute node for the created instance:</p> <ul style="list-style-type: none"> • OS-EXT-SRV-ATTR:host • OS-EXT-SRV-ATTR:hypervisor_hostname |
| Verify network connectivity of a virtual machine instance without an assigned floating IP address. | <p>You must be able to successfully execute the following actions:</p> <ul style="list-style-type: none"> • Ping the internal IP address of the instance. • Establish a TCP connection between the controller node and the internal IP address of the instance. |

| | |
|---|---|
| <p>Verify network connectivity of an instance with an assigned floating IP address.</p> | <p>You must be able to successfully execute the following actions:</p> <ul style="list-style-type: none"> • Ping the floating IP address of the instance. • Establish a TCP connection between the controller node and the floating IP address of the instance. • Access the Internet from the instance. |
| <p>Verify that an instance can access metadata.</p> | <p>Instance must be able to receive metadata, such as a public key.</p> |

Note

In the examples provided in this section, the instance uses the default CirrOS image.

To validate the deployment:

1. Log in to a controller node.
2. Verify the state of the OpenStack services:
 1. View the list of hypervisors:

```
nova hypervisor-list
```

Example of correct output:

```
+-----+-----+-----+
| Hypervisor hostname | State | Status |
+-----+-----+-----+
| rhel-node.domain.tld | up   | enabled |
+-----+-----+-----+
```

2. View the list of services:

```
nova service-list
```

Example of correct output:

```
+-----+-----+-----+-----+-----+
| Binary      | Host           | Zone   | Status | State|
+-----+-----+-----+-----+-----+
| nova-compute | rhel-node.domain.tld | nova   | enabled | up  |
+-----+-----+-----+-----+-----+
```

3. View the list of neutron agents (no DVR):

```
neutron agent-list
```

Example of correct output:

```
+-----+-----+-----+-----+
| agent_type | host | alive | admin_state_up |
+-----+-----+-----+-----+
| Open vSwitch agent | rhel-node.domain.tld | :- ) | True |
```

4. View the list of neutron agents (with DVR):

```
neutron agent-list
```

Example of correct output:

```
+-----+-----+-----+-----+
| agent_type | host | alive | admin_state_up |
+-----+-----+-----+-----+
| Metadata agent | rhel-node.domain.tld | :- ) | True |
| Open vSwitch agent | rhel-node.domain.tld | :- ) | True |
| L3 agent | rhel-node.domain.tld | :- ) | True |
```

3. Launch a virtual machine instance with the Red Hat family distributions availability zone.

Example:

```
nova boot --flavor m1.tiny --image TestVM --nic
net-id=9c32fd49-f680-4aef-b012-d8750f9d0458 --security-group
default --availability-zone nova:rhel-node.domain.tld test-instance
```

4. View the list of virtual machines:

```
nova list
```

Example of correct output:

```
+-----+-----+-----+-----+-----+
| Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+-----+
| test-instance | ACTIVE | - | Running | net04=192.168.111.6 |
```

5. View the detailed information about the virtual machine:

```
nova show test-instance
```

Example of the correct output:

```
+-----+-----+
| Property          | Value                |
+-----+-----+
| OS-DCF:diskConfig | MANUAL               |
| OS-EXT-AZ:availability_zone | nova                |
| OS-EXT-SRV-ATTR:host | rhel-node.domain.tld |
| OS-EXT-SRV-ATTR:hypervisor_hostname | rhel-node.domain.tld |
| OS-EXT-SRV-ATTR:instance_name | instance-00000004   |
| OS-EXT-STS:power_state | 1                   |
| OS-EXT-STS:task_state | -                   |
| OS-EXT-STS:vm_state | active              |
| OS-SRV-USG:launched_at | 2015-12-02T07:03:42.000000 |
| OS-SRV-USG:terminated_at | -                   |
| accessIPv4        |                      |
| accessIPv6        |                      |
| net04 network     | 192.168.111.6      |
| config_drive      |                      |
| created           | 2015-12-02T09:25:18Z |
| flavor            | m1.tiny (1)         |
| hostId            | 0e8fb55c2ae5be7be7e99c27998234 |
| id                | 310e554f-0579-4dd1-8802-7a592c |
| image             | TestVM (5feb2351-2f4f-4c2c-bdd |
| key_name          | -                   |
| metadata          | {}                  |
| name              | test-instance      |
| os-extended-volumes:volumes_attached | []                  |
| progress          | 0                   |
| security_groups   | default            |
| status            | ACTIVE              |
| tenant_id         | a9098718a0004bb39ed5035a9b2b53 |
| updated           | 2015-12-02T09:25:32Z |
| user_id           | 1891425ba6834df0aac76478bc44b6 |
+-----+-----+
```

6. Verify network connectivity of the virtual machine instance without a floating IP address:

1. Log in to the controller node.
2. Determine the qrouter namespace ID:

```
#ip netns
```

Example of output:

```
qrouter-4cb24468-fa30-446d-a93b-122070360dce
qdhcp-9c32fd49-f680-4aef-b012-d8750f9d0458
haproxy
vrouter
```

3. Ping the internal IP of an instance from the qrouter network namespace.

Example:

```
ip netns exec qrouter-4cb24468-fa30-446d-a93b-122070360dce
ping 192.168.111.6
```

System response:

```
PING 192.168.111.6 (192.168.111.6) 56(84) bytes of data.
64 bytes from 192.168.111.6: icmp_seq=1 ttl=64 time=3.59 ms
64 bytes from 192.168.111.6: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 192.168.111.6: icmp_seq=3 ttl=64 time=0.802 ms
```

4. Enable the SSH access to the virtual machine:

```
neutron security-group-rule-create --direction ingress
--remote-ip-prefix 0.0.0.0/0 --port-range-min 22 --port-range-max
22 --protocol tcp default
```

5. Verify the SSH connection to the virtual machine instance using SSH:

```
ip netns exec qrouter-4cb24468-fa30-446d-a93b-122070360dce nc
192.168.111.6 22 -vv
Connection to 192.168.111.6 22 port [tcp/ssh] succeeded!
SSH-2.0-dropbear_2012.55
```

7. Verify network connectivity of the virtual machine instance with a floating IP address:

1. Verify if any floating IP addresses are available:

```
neutron floatingip-list
```

2. If no floating IP addresses are available, create a floating IP address:

```
neutron floatingip-create --floating-ip-address IP_ADDRESS
FLOATING_NETWORK
```

3. Assign a floating IP address to the instance:

```
neutron floatingip-associate FLOATINGIP_ID PORT
```

4. Enable access using SSH to the virtual machine instance:


```
neutron security-group-rule-create --direction ingress
--remote-ip-prefix 0.0.0.0/0 --port-range-min 22 --port-range-max 22
--protocol tcp default
```

5. Ping the floating IP address of the instance.

6. Verify the SSH connection:

Example:

```
nc 172.16.164.244 22 -vv
```

System Response:

```
Connection to 172.16.164.244 22 port [tcp/ssh] succeeded!
SSH-2.0-dropbear_2012.55
```

8. Log in to the virtual machine instance.

9. Verify Internet connectivity:

Example:

```
ssh cirros@172.16.164.244 curl docs.openstack.org
```

10. Verify that the instance can access metadata:

1. Log in to the controller node.

2. Generate an SSH key pair:

```
ssh-keygen -q -f ssh-key
```

3. Add the SSH key pair to nova:

Example:

```
nova keypair-add novakey --pub-key ssh-key.pub
```

4. Launch a virtual machine with the SSH key:

Example:

```
nova boot --flavor m1.tiny --image TestVM --nic
net-id=27fdca6e-71a5-4eb6-b728-f39902ffe63d --security-group default
--availability-zone nova:rhel-1.domain.tld --key-name novakey testvm
```

5. Log in to the instance using the generated SSH key.

Example:

```
ip netns exec qrouter-80c5841c-7bc4-4718-bf95-5a5a5b81b0b9 ssh -i
ssh-key cirros@192.168.111.5
```

You must be able to log in without a password.

6. Verify that you can access the metadata folder from the instance:

Example:

```
curl 169.254.169.254/latest/meta-data
```

System Response:

```
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
hostname  
instance-action  
instance-id  
instance-type  
local-hostname  
local-ipv4  
placement/  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups
```

11 Reboot the compute node.

.

12 Repeat step 1 - step 10.

.

Segregate workloads

To ensure the workloads run on the compute nodes with Red Hat family distributions, you can separate them from the Ubuntu workloads using availability zones and host aggregates.

To segregate workloads:

1. Log in to the Horizon dashboard.
2. Create host aggregates and availability zones for the compute nodes and for the Ubuntu nodes:
 1. In the left-hand panel, click Admin > System > Host Aggregates.
 2. Click Create Host Aggregate.
 3. In the Host Aggregation Information tab, fill the Name and the Availability Zone fields with descriptive names.
 4. In the Manage Hosts within Aggregate add corresponding nodes.
 5. Click Create Host Aggregate.

When later you launch new virtual machines in Horizon or using CLI, assign a corresponding availability zone for these machines.

Example of an environment with segregation

The following table provides an example of an environment with segregation:

Example of an environment with segregation

| Item | Description |
|--------------------------------------|---|
| OpenStack Environment configuration: | <ul style="list-style-type: none"> • 2 Ubuntu nodes: node-11.domain.tld, node-12.domain.tld* • 2 compute nodes with Red Hat family distribution: rhel-1.domain.tld, rhel-2.domain.tld • 1 Ubuntu controller node: node-10.domain.tld |
| Host aggregates: | <ul style="list-style-type: none"> • RHEL_compute_nodes • Ubuntu_compute_nodes |
| Availability zones: | <ul style="list-style-type: none"> • RHEL7 • Ubuntu |

The following image is a screenshot of the above configuration in Horizon.

Host Aggregates

Filter

| <input type="checkbox"/> | Name | Availability Zone | Hosts | Metadata |
|--------------------------|----------------------|-------------------|--|----------------------------|
| <input type="checkbox"/> | RHEL7_compute_nodes | RHEL7 | rhel-1.domain.tld rhel-2.domain.tld | availability_zone = RHEL7 |
| <input type="checkbox"/> | Ubuntu_compute_nodes | Ubuntu | node-11.domain.tld node-12.domain.tld | availability_zone = Ubuntu |

Displaying 2 items

Availability Zones

| Availability Zone Name | Hosts |
|------------------------|--|
| internal | node-10.domain.tld (Services Up) |
| RHEL7 | rhel-1.domain.tld (Services Up) rhel-2.domain.tld (Services Up) |
| Ubuntu | node-12.domain.tld (Services Up) node-11.domain.tld (Services Up) |

Displaying 3 items