**Neutron / ML2 Configuration**

Version 1.0

**Nexus / ML2 Configuration**

**ATTENTION**

The information contained in this guide is for training purposes only. This guide contains information and activities that, while beneficial for purposes of training in close, non-production environment, can result in downtime or other severe consequences and therefore are not intended as a reference guide.

No part of this book may be reproduced in any form or by any means (graphic, electronic, mechanical, including photocopying, recording or taping, or storage in an electronic retrieval system) without permission of OneCloud Consulting.

OneCloud Consulting reserves the right to change the contents herein without any prior notice.

[Nexus / ML2 Configuration 4](#_Toc389757275)

[Overview 4](#_Toc389757276)

[Assumptions 4](#_Toc389757277)

[Neutron / ML2 Configuration 5](#_Toc389757278)

[Configure Controller Node 5](#_Toc389757279)

[Configure Network Node 9](#_Toc389757280)

[Configure compute node 14](#_Toc389757281)

[Nexus Configuration 18](#_Toc389757282)

[VLAN Creation 19](#_Toc389757283)

Nexus / ML2 Configuration

# Overview

The Modular Layer 2 (ml2) plugin is a framework allowing OpenStack Networking to simultaneously utilize the variety of layer 2 networking technologies found in complex real-world data centers. It currently works with the existing openvswitch, linuxbridge, and hyperv L2 agents, and is intended to replace and deprecate the monolithic plugins associated with those L2 agents. The ml2 framework is also intended to greatly simplify adding support for new L2 networking technologies, requiring much less initial and ongoing effort than would be required to add a new monolithic core plugin.

# Assumptions

* OpenStack (Keystone, Glance and Nova) services are up and running

Neutron / ML2 Configuration

# Configure Controller Node

Before you configure OpenStack Networking (neutron), you must create a database and Identity service credentials including a user and service.

1. Connect to the database as the root user, create the neutron database, and grant the proper access to it:

Replace NEUTRON\_DBPASS with a suitable password.

$ mysql -u root -p

mysql> CREATE DATABASE neutron;

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' \

IDENTIFIED BY '*NEUTRON\_DBPASS*';

mysql> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' \

IDENTIFIED BY '*NEUTRON\_DBPASS*';

1. Create Identity service credentials for Networking:
2. Create the neutron user:

Replace *NEUTRON\_PASS* with a suitable password and *neutron@example.com* with a suitable e-mail address.

$ keystone user-create --name neutron --pass *NEUTRON\_PASS* --email *neutron@example.com*

1. Link the neutron user to the service tenant and admin role:

$ keystone user-role-add --user neutron --tenant service --role admin

1. Create the neutron service:

$ keystone service-create --name neutron --type network --description "OpenStack Networking"

1. Create the service endpoint:

$ keystone endpoint-create \

--service-id $(keystone service-list | awk '/ network / {print $2}') \

--publicurl http://*controller*:9696 \

--adminurl http://*controller*:9696 \

--internalurl http://*controller*:9696

**To install the Networking components**

# apt-get install neutron-server neutron-plugin-ml2

**To configure the Networking server component**

The Networking server component configuration includes the database, authentication mechanism, message broker, topology change notifier, and plug-in.

1. Configure Networking to use the database:
2. Edit the /etc/neutron/neutron.conf file and add the following key to the [database] section:

Replace NEUTRON\_DBPASS with the password you chose for the database.

[database]

...

connection = mysql://neutron:NEUTRON\_DBPASS@controller/neutron

1. Configure Networking to use the Identity service for authentication:
2. Edit the /etc/neutron/neutron.conf file and add the following key to the [DEFAULT] section:

[DEFAULT]

...

auth\_strategy = keystone

Add the following keys to the [keystone\_authtoken] section:

Replace *NEUTRON\_PASS* with the password you chose for the neutron user in the Identity service.

[keystone\_authtoken]

...

auth\_uri = http://controller:5000

auth\_host = controller

auth\_protocol = http

auth\_port = 35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

1. Configure Networking to use the message broker:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

Replace RABBIT\_PASS with the password you chose for the guest account in RabbitMQ

[DEFAULT]

...

rpc\_backend = neutron.openstack.common.rpc.impl\_kombu

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

1. Configure Networking to notify Compute about network topology changes:

Replace SERVICE\_TENANT\_ID with the service tenant identifier (id) in the Identity service and NOVA\_PASS with the password you chose for thenova user in the Identity service.

1. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

notify\_nova\_on\_port\_status\_changes = True

notify\_nova\_on\_port\_data\_changes = True

nova\_url = http://controller:8774/v2

nova\_admin\_username = nova

nova\_admin\_tenant\_id = SERVICE\_TENANT\_ID

nova\_admin\_password = NOVA\_PASS

nova\_admin\_auth\_url = http://controller:35357/v2.0

To obtain the service tenant identifier (id):

$ source admin-openrc.sh

$ keystone tenant-get service

+-------------+----------------------------------+

| Property | Value |

+-------------+----------------------------------+

| description | Service Tenant |

| enabled | True |

| id | f727b5ec2ceb4d71bad86dfc414449bf |

| name | service |

+-------------+----------------------------------+

1. Configure Networking to use the Modular Layer 2 (ML2) plug-in and associated services:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

1. Comment out any lines in the [service\_providers] section.

**To configure the Modular Layer 2 (ML2) plug-in**

The ML2 plug-in uses the Open vSwitch (OVS) mechanism (agent) to build the virtual networking framework for instances. However, the controller node does not need the OVS agent or service because it does not handle instance network traffic.

1. Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file:

Add the following keys to the [ml2] section:

[ml2]

...

type\_drivers = vlan

tenant\_network\_types = vlan

mechanism\_drivers = openvswitch

Add the following key to the [ml2\_type\_vlan] section:

[ml2\_type\_vlan]

...

tunnel\_id\_ranges = 1:1000

Add the [securitygroup] section and the following keys to it:

[securitygroup]

...

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

enable\_security\_group = True

**To configure Compute to use Networking**

By default, most distributions configure Compute to use legacy networking. You must reconfigure Compute to manage networks through Networking.

1. Edit the /etc/nova/nova.conf and add the following keys to the [DEFAULT] section:

Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

[DEFAULT]

...

network\_api\_class = nova.network.neutronv2.api.API

neutron\_url = http://controller:9696

neutron\_auth\_strategy = keystone

neutron\_admin\_tenant\_name = service

neutron\_admin\_username = neutron

neutron\_admin\_password = NEUTRON\_PASS

neutron\_admin\_auth\_url = http://controller:35357/v2.0

linuxnet\_interface\_driver = nova.network.linux\_net.LinuxOVSInterfaceDriver

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

security\_group\_api = neutron

**To finalize installation**

1. Restart the Compute services:

# service nova-api restart

# service nova-scheduler restart

# service nova-conductor restart

1. Restart the Networking service:

# service neutron-server restart

# Configure Network Node

Before you configure OpenStack Networking, you must enable certain kernel networking functions.

1. Edit /etc/sysctl.conf to contain the following:

net.ipv4.ip\_forward=1

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

1. Implement the changes:

# sysctl -p

**To install the Networking components**

# apt-get install neutron-plugin-ml2 neutron-plugin-openvswitch-agent openvswitch-datapath-dkms \

neutron-l3-agent neutron-dhcp-agent

**To configure the Networking common components**

The Networking common component configuration includes the authentication mechanism, message broker, and plug-in.

1. Configure Networking to use the Identity service for authentication:
2. Edit the /etc/neutron/neutron.conf file and add the following key to the [DEFAULT] section:

[DEFAULT]

...

auth\_strategy = keystone

Add the following keys to the [keystone\_authtoken] section:

Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

[keystone\_authtoken]

...

auth\_uri = http://controller:5000

auth\_host = controller

auth\_protocol = http

auth\_port = 35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

1. Configure Networking to use the message broker:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

Replace RABBIT\_PASS with the password you chose for the guest account in RabbitMQ.

[DEFAULT]

...

rpc\_backend = neutron.openstack.common.rpc.impl\_kombu

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

1. Configure Networking to use the Modular Layer 2 (ML2) plug-in and associated services:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

1. Comment out any lines in the [service\_providers] section.

**To configure the Layer-3 (L3) agent**

The [Layer-3 (L3) agent](http://docs.openstack.org/trunk/install-guide/install/apt/content/neutron-ml2-network-node.html) provides routing services for instance virtual networks.

1. Edit the /etc/neutron/l3\_agent.ini file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver

use\_namespaces = True

**To configure the DHCP agent**

The [DHCP agent](http://docs.openstack.org/trunk/install-guide/install/apt/content/neutron-ml2-network-node.html) provides [DHCP](http://docs.openstack.org/trunk/install-guide/install/apt/content/neutron-ml2-network-node.html) services for instance virtual networks.

1. Edit the /etc/neutron/dhcp\_agent.ini file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver

dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq

use\_namespaces = True

**To configure the metadata agent**

The [metadata agent](http://docs.openstack.org/trunk/install-guide/install/apt/content/neutron-ml2-network-node.html) provides configuration information such as credentials for remote access to instances.

1. Edit the /etc/neutron/metadata\_agent.ini file and add the following keys to the [DEFAULT] section:

Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service. Replace METADATA\_SECRET with a suitable secret for the metadata proxy.

[DEFAULT]

...

auth\_url = http://controller:5000/v2.0

auth\_region = regionOne

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

nova\_metadata\_ip = controller

metadata\_proxy\_shared\_secret = METADATA\_SECRET

1. On the controller node, edit the /etc/nova/nova.conf file and add the following keys to the [DEFAULT] section:

Replace METADATA\_SECRET with the secret you chose for the metadata proxy.

[DEFAULT]

...

service\_neutron\_metadata\_proxy = true

neutron\_metadata\_proxy\_shared\_secret = METADATA\_SECRET

1. On the *controller* node, restart the Compute [API](http://docs.openstack.org/trunk/install-guide/install/apt/content/neutron-ml2-network-node.html) service:

# service nova-api restart

**To configure the Modular Layer 2 (ML2) plug-in**

The ML2 plug-in uses the Open vSwitch (OVS) mechanism (agent) to build virtual networking framework for instances.

* Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file.

Add the following keys to the [ml2] section:

[ml2]

...

type\_drivers = vlan

tenant\_network\_types = vlan

mechanism\_drivers = openvswitch

Add the following keys to the [ml2\_type\_vlan] section:

[ml2\_type\_vlan]

...

tunnel\_id\_ranges = 1:1000

Add the [ovs] section and the following keys to it:

Replace INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS with the IP address of the instance tunnels network interface on your network node.

[ovs]

...

local\_ip = INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS

tunnel\_type = vlan

enable\_tunneling = True

Add the [securitygroup] section and the following keys to it:

[securitygroup]

...

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

enable\_security\_group = True

**To configure the Open vSwitch (OVS) service**

The OVS service provides the underlying virtual networking framework for instances. The integration bridge br-int handles internal instance network traffic within OVS. The external bridge br-ex handles external instance network traffic within OVS. The external bridge requires a port on the physical external network interface to provide instances with external network access. In essence, this port bridges the virtual and physical external networks in your environment.

1. Restart the OVS service:

# service openvswitch-switch restart

1. Add the integration bridge:

# ovs-vsctl add-br br-int

1. Add the external bridge:

# ovs-vsctl add-br br-ex

1. Add a port to the external bridge that connects to the physical external network interface:

Replace *INTERFACE\_NAME* with the actual interface name. For example, *eth2* or *ens256*.

# ovs-vsctl add-port br-ex *INTERFACE\_NAME*

**To finalize the installation**

* Restart the Networking services:

# service neutron-plugin-openvswitch-agent restart

# service neutron-l3-agent restart

# service neutron-dhcp-agent restart

# service neutron-metadata-agent restart

# Configure compute node

Before you configure OpenStack Networking, you must enable certain kernel networking functions.

1. Edit /etc/sysctl.conf to contain the following:

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

1. Implement the changes:

# sysctl -p

**To install the Networking components**

* # apt-get install neutron-common neutron-plugin-ml2 neutron-plugin-openvswitch-agent \
* openvswitch-datapath-dkms

**To configure the Networking common components**

The Networking common component configuration includes the authentication mechanism, message broker, and plug-in.

1. Configure Networking to use the Identity service for authentication:
2. Edit the /etc/neutron/neutron.conf file and add the following key to the [DEFAULT] section:

[DEFAULT]

...

auth\_strategy = keystone

Add the following keys to the [keystone\_authtoken] section:

Replace NEUTRON\_PASS with the password you chose for the neutron user in the Identity service.

[keystone\_authtoken]

...

auth\_uri = http://controller:5000

auth\_host = controller

auth\_protocol = http

auth\_port = 35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

1. Configure Networking to use the message broker:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

Replace *RABBIT\_PASS* with the password you chose for the guest account in RabbitMQ.

[DEFAULT]

...

rpc\_backend = neutron.openstack.common.rpc.impl\_kombu

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

1. Configure Networking to use the Modular Layer 2 (ML2) plug-in and associated services:
2. Edit the /etc/neutron/neutron.conf file and add the following keys to the [DEFAULT] section:

[DEFAULT]

...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

1. Comment out any lines in the [service\_providers] section.

**To configure the Modular Layer 2 (ML2) plug-in**

The ML2 plug-in uses the Open vSwitch (OVS) mechanism (agent) to build the virtual networking framework for instances.

* Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file:

Add the following keys to the [ml2] section:

[ml2]

...

type\_drivers = vlan

tenant\_network\_types = vlan

mechanism\_drivers = openvswitch

Add the following keys to the [ml2\_type\_gre] section:

[ml2\_type\_vlan]

...

tunnel\_id\_ranges = 1:1000

Add the [ovs] section and the following keys to it:

Replace INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS with the IP address of the instance tunnels network interface on your compute node.

[ovs]

...

local\_ip = INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS

tunnel\_type = vlan

enable\_tunneling = True

Add the [securitygroup] section and the following keys to it:

[securitygroup]

...

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

enable\_security\_group = True

**To configure the Open vSwitch (OVS) service**

The OVS service provides the underlying virtual networking framework for instances. The integration bridge br-int handles internal instance network traffic within OVS.

1. Restart the OVS service:

# service openvswitch-switch restart

1. Add the integration bridge:

# ovs-vsctl add-br br-int

**To configure Compute to use Networking**

By default, most distributions configure Compute to use legacy networking. You must reconfigure Compute to manage networks through Networking.

* Edit the /etc/nova/nova.conf and add the following keys to the [DEFAULT] section:

Replace *NEUTRON\_PASS* with the password you chose for the neutron user in the Identity service.

[DEFAULT]

...

network\_api\_class = nova.network.neutronv2.api.API

neutron\_url = http://controller:9696

neutron\_auth\_strategy = keystone

neutron\_admin\_tenant\_name = service

neutron\_admin\_username = neutron

neutron\_admin\_password = NEUTRON\_PASS

neutron\_admin\_auth\_url = http://controller:35357/v2.0

linuxnet\_interface\_driver = nova.network.linux\_net.LinuxOVSInterfaceDriver

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

security\_group\_api = neutron

**To finalize the installation**

1. Restart the Compute service:

# service nova-compute restart

1. Restart the Open vSwitch (OVS) agent:

# service neutron-plugin-openvswitch-agent restart

# Nexus Configuration

**Installation of ncclient**

We are facing “Unknown Host Key” issue with “ncclient-0.4.0” version. So install the latest version of ncclient which is “[ncclient-0.4.1](https://pypi.python.org/packages/source/n/ncclient/ncclient-0.4.1.tar.gz#md5=1c7cac36ad625b0a9d1ab8f94c31daa4)”.

**Download the package:**

wget https://pypi.python.org/packages/source/n/ncclient/ncclient-0.4.1.tar.gz#md5=1c7cac36ad625b0a9d1ab8f94c31daa4

or **Install using pip:**

pip install ncclient==0.4.1

This may ask several dependencies, install them using apt-get.

**Directory Structure**

The Cisco Nexus mechanism driver code can be found here:

neutron/plugins/ml2/drivers/cisco/nexus

The Cisco Nexus mechanism configuration file code is located at:

neutron/plugins/ml2/ml2\_conf\_cisco.ini

**Configuration**

Using the syntax template found here, etc/neutron/plugins/ml2/ml2\_conf\_cisco.ini, add the Nexus switch information to a configuration file included on the command line when the neutron-server is started. The format should include the IP address of the switch, a host that's connected to the switch and the port on the switch that host is connected to. Also, add the Nexus switch credential username and password. You can configure multiple switches as well as multiple hosts per switch as shown in the example below:

# Use section header 'ml2\_mech\_cisco\_nexus:' followed by the IP address of the Nexus switch.

[ml2\_mech\_cisco\_nexus:1.1.1.1]

# Hostname and port used on the switch for this compute host.

# Where 1/2 indicates the "interface ethernet 1/2" port on the switch.

compute-1=1/2

# Port number where the SSH will be running at the Nexus Switch. Default is 22 so this variable

# only needs to be configured if different.

# ssh\_port=22

# Provide the Nexus log in information

username=admin

password=mySecretPasswordForNexus

**Configure mechanism drivers**

MECHANISM\_DRIVERS=openvswitch,cisco\_nexus

# VLAN Creation

**Creating Network:**

neutron net-create vlan1005 --provider:network\_type vlan --provider:physical\_network physnet1 --provider:segmentation\_id 1005 --shared --router:external=True

**Creating Subnet and attaching to created network:**

neutron subnet-create --name subnet1005 --allocation-pool start=192.168.250.10,end=192.168.250.254 vlan1005 192.168.250.0/24 --dns\_nameservers list=true 10.121.12.10

**Verify the creation of VLAN network using:**

neutron net-list

Thank you

