

ML2 plug-in

[«](#) (config-services-agent.html) [»](#) (config-address-scopes.html) [%0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A-----%0ARelease:%0A2012.0.1.dev11%0Aon%0A202018-03-07%0A2021:05%0ASHA:%0A2043df2709acbdce86686a40b75fd34e96880427d0%0ASource:%0Ahttps://git.openstack.org/cgiit/openstack/neutron/tree/doc/source/admin/config-mL2.rst%0AURL: https://docs.openstack.org/neutron/queens/admin/config-mL2.html&field.tags=doc\)](https://bugs.launchpad.net/neutron/+filebug?field.title=ML2%20plugin%20in%20Neutron&field.comment=%0A%0A%0AThis+bug+tracker+is+for+errors+with+the+documentation,+use+the+following+as+a+template+and+remove+or+add+fields+as+you+see+fit.+Convert+%5B+%5D+into+%5Bx%5D+to+check+boxes:%0A%0A-+%5B+%5D+This+doc+is+inaccurate+in+this+way:+_____%0A-+%5B+%5D+This+is+a+doc+addition+request.%0A-+%5B+%5D+I+have+a+fix+to+the+document+that+I+can+paste+below+including+example:+input+and+output.)

UPDATED: 2018-03-07 21:05

Architecture

The Modular Layer 2 (ML2) neutron plug-in is a framework allowing OpenStack Networking to simultaneously use the variety of layer 2 networking technologies found in complex real-world data centers. The ML2 framework distinguishes between the two kinds of drivers that can be configured:

- Type drivers

Define how an OpenStack network is technically realized. Example: VXLAN

Each available network type is managed by an ML2 type driver. Type drivers maintain any needed type-specific network state. They validate the type specific information for provider networks and are responsible for the allocation of a free segment in project networks.

- Mechanism drivers

Define the mechanism to access an OpenStack network of a certain type. Example: Open vSwitch mechanism driver.

The mechanism driver is responsible for taking the information established by the type driver and ensuring that it is properly applied given the specific networking mechanisms that have been enabled.

Mechanism drivers can utilize L2 agents (via RPC) and/or interact directly with external devices or controllers.

Multiple mechanism and type drivers can be used simultaneously to access different ports of the same virtual network.

Todo

Picture showing relationships

ML2 driver support matrix¶

Mechanism drivers and L2 agents

type driver / mech driver	Flat	VLAN	VXLAN	GRE
Open vSwitch	yes	yes	yes	yes
Linux bridge	yes	yes	yes	no
SRIOV	yes	yes	no	no
MacVTap	yes	yes	no	no
L2 population	no	no	yes	yes

✔ Note

L2 population is a special mechanism driver that optimizes BUM (Broadcast, unknown destination address, multicast) traffic in the overlay networks VXLAN and GRE. It needs to be used in conjunction with either the Linux bridge or the Open vSwitch mechanism driver and cannot be used as standalone mechanism driver. For more information, see the *Mechanism drivers* section below.

Configuration

Network type drivers¶

To enable type drivers in the ML2 plug-in, edit the `/etc/neutron/plugins/ml2/ml2_conf.ini` file:

```
[m12]
type drivers = flat,vlan,vxlan,gre
```

✔ **Note**

For more details , see the [Bug 1567792 \(https://bugs.launchpad.net/openstack-manuals/+bug/1567792\)](https://bugs.launchpad.net/openstack-manuals/+bug/1567792).

For more details, see the Networking configuration options ([../configuration/m12-conf.html](#)) of Configuration Reference.

The following type drivers are available

- Flat
- VLAN
- GRE
- VXLAN

Provider network types[¶]

Provider networks provide connectivity like project networks. But only administrative (privileged) users can manage those networks because they interface with the physical network infrastructure. More information about provider networks see [OpenStack Networking \(intro-os-networking.html\)](#).

- Flat

The administrator needs to configure a list of physical network names that can be used for provider networks. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#ml2-type-flat\)](#).

- VLAN

The administrator needs to configure a list of physical network names that can be used for provider networks. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#ml2-type-vlan\)](#).

- GRE

No additional configuration required.

- VXLAN

The administrator can configure the VXLAN multicast group that should be used.

Note

VXLAN multicast group configuration is not applicable for the Open vSwitch agent.

As of today it is not used in the Linux bridge agent. The Linux bridge agent has its own agent specific configuration option. For more details, see the [Bug 1523614 \(https://bugs.launchpad.net/neutron/+bug/1523614\)](#).

Project network types[¶]

Project networks provide connectivity to instances for a particular project. Regular (non-privileged) users can manage project networks within the allocation that an administrator or operator defines for them. More information about project and provider networks see [OpenStack Networking \(intro-os-networking.html\)](#).

Project network configurations are made in the `/etc/neutron/plugins/ml2/ml2_conf.ini` configuration file on the neutron server:

- VLAN

The administrator needs to configure the range of VLAN IDs that can be used for project network allocation. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#ml2-type-vlan\)](#).

- GRE

The administrator needs to configure the range of tunnel IDs that can be used for project network allocation. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#ml2-type-gre\)](#).

- VXLAN

The administrator needs to configure the range of VXLAN IDs that can be used for project network allocation. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#ml2-type-vxlan\)](#).

Note

Flat networks for project allocation are not supported. They only can exist as a provider network.

Mechanism drivers[¶]

To enable mechanism drivers in the ML2 plug-in, edit the `/etc/neutron/plugins/ml2/ml2_conf.ini` file on the neutron server:

```
[ml2]
mechanism_drivers = ovs,l2pop
```

Note

For more details, see the [Bug 1567792 \(https://bugs.launchpad.net/openstack-manuals/+bug/1567792\)](#).

For more details, see the [Configuration Reference \(./configuration/ml2-conf.html#ml2\)](#).

- Linux bridge

No additional configurations required for the mechanism driver. Additional agent configuration is required. For details, see the related *L2 agent* section below.

- Open vSwitch

No additional configurations required for the mechanism driver. Additional agent configuration is required. For details, see the related *L2 agent* section below.

- SRIOV

The SRIOV driver accepts all PCI vendor devices.

- MacVTap

No additional configurations required for the mechanism driver. Additional agent configuration is required. Please see the related section.

- L2 population

The administrator can configure some optional configuration options. For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#l2pop\)](#).

- Specialized

- Open source

External open source mechanism drivers exist as well as the neutron integrated reference implementations. Configuration of those drivers is not part of this document. For example:

- OpenDaylight
- OpenContrail

- Proprietary (vendor)

External mechanism drivers from various vendors exist as well as the neutron integrated reference implementations.

Configuration of those drivers is not part of this document.

Extension Drivers

The ML2 plug-in also supports extension drivers that allows other pluggable drivers to extend the core resources implemented in the ML2 plug-in (**networks, ports**, etc.). Examples of extension drivers include support for QoS, port security, etc. For more details see the **extension_drivers** configuration option in the [Configuration Reference \(./configuration/ml2-conf.html#ml2.extension_drivers\)](#).

Agents

L2 agent

An L2 agent serves layer 2 (Ethernet) network connectivity to OpenStack resources. It typically runs on each Network Node and on each Compute Node.

- Open vSwitch agent

The Open vSwitch agent configures the Open vSwitch to realize L2 networks for OpenStack resources.

Configuration for the Open vSwitch agent is typically done in the **openvswitch_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/openvswitch-agent.html\)](#).

- Linux bridge agent

The Linux bridge agent configures Linux bridges to realize L2 networks for OpenStack resources.

Configuration for the Linux bridge agent is typically done in the **linuxbridge_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/linuxbridge-agent.html\)](#).

- SRIOV Nic Switch agent

The sriov nic switch agent configures PCI virtual functions to realize L2 networks for OpenStack instances. Network attachments for other resources like routers, DHCP, and so on are not supported.

Configuration for the SRIOV nic switch agent is typically done in the **sriov_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/sriov-agent.html\)](#).

- MacVTap agent

The MacVTap agent uses kernel MacVTap devices for realizing L2 networks for OpenStack instances. Network attachments for other resources like routers, DHCP, and so on are not supported.

Configuration for the MacVTap agent is typically done in the **macvtap_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/macvtap-agent.html\)](#).

L3 agent

The L3 agent offers advanced layer 3 services, like virtual Routers and Floating IPs. It requires an L2 agent running in parallel.

Configuration for the L3 agent is typically done in the **l3_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/l3-agent.html\)](#).

DHCP agent

The DHCP agent is responsible for DHCP (Dynamic Host Configuration Protocol) and RADVD (Router Advertisement Daemon) services. It requires a running L2 agent on the same node.

Configuration for the DHCP agent is typically done in the **dhcp_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/dhcp-agent.html\)](#).

Metadata agent

The Metadata agent allows instances to access cloud-init meta data and user data via the network. It requires a running L2 agent on the same node.

Configuration for the Metadata agent is typically done in the **metadata_agent.ini** configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/metadata-agent.html\)](#).

L3 metering agent¶

The L3 metering agent enables layer3 traffic metering. It requires a running L3 agent on the same node.

Configuration for the L3 metering agent is typically done in the `metering_agent.ini` configuration file. Make sure that on agent start you pass this configuration file as argument.

For a detailed list of configuration options, see the related section in the [Configuration Reference \(./configuration/metering-agent.html\)](#).

Security¶

L2 agents support some important security configurations.

- Security Groups
For more details, see the related section in the [Configuration Reference \(./configuration/ml2-conf.html#securitygroup\)](#).
- Arp Spoofing Prevention
Configured in the *L2 agent* configuration.

Reference implementations¶

Overview¶

In this section, the combination of a mechanism driver and an L2 agent is called 'reference implementation'. The following table lists these implementations:

Mechanism drivers and L2 agents¶

Mechanism Driver	L2 agent
Open vSwitch	Open vSwitch agent
Linux bridge	Linux bridge agent
SRIOV	SRIOV nic switch agent
MacVTap	MacVTap agent
L2 population	Open vSwitch agent, Linux bridge agent

The following tables shows which reference implementations support which non-L2 neutron agents:

Reference implementations and other agents¶

Reference Implementation	L3 agent	DHCP agent	Metadata agent	L3 Metering agent
Open vSwitch & Open vSwitch agent	yes	yes	yes	yes
Linux bridge & Linux bridge agent	yes	yes	yes	yes
SRIOV & SRIOV nic switch agent	no	no	no	no
MacVTap & MacVTap agent	no	no	no	no

● Note

L2 population is not listed here, as it is not a standalone mechanism. If other agents are supported depends on the conjunctive mechanism driver that is used for binding a port.

More information about L2 population see the [OpenStack Manuals \(http://docs.ocselected.org/openstack-manuals/kilo/networking-guide/content/ml2_l2pop_scenarios.html\)](http://docs.ocselected.org/openstack-manuals/kilo/networking-guide/content/ml2_l2pop_scenarios.html).

Buying guide¶

This guide characterizes the L2 reference implementations that currently exist.

- Open vSwitch mechanism and Open vSwitch agent
Can be used for instance network attachments as well as for attachments of other network resources like routers, DHCP, and so on.
- Linux bridge mechanism and Linux bridge agent
Can be used for instance network attachments as well as for attachments of other network resources like routers, DHCP, and so on.
- SRIOV mechanism driver and SRIOV NIC switch agent
Can only be used for instance network attachments (device_owner = compute).
Is deployed besides an other mechanism driver and L2 agent such as OVS or Linux bridge. It offers instances direct access to the network adapter through a PCI Virtual Function (VF). This gives an instance direct access to hardware capabilities and high performance networking.
The cloud consumer can decide via the neutron APIs VNIC_TYPE attribute, if an instance gets a normal OVS port or an SRIOV port.
Due to direct connection, some features are not available when using SRIOV. For example, DVR, security groups, migration.
For more information see the [SR-IOV \(config-sriov.html#config-sriov\)](#).
- MacVTap mechanism driver and MacVTap agent
Can only be used for instance network attachments (device_owner = compute) and not for attachment of other resources like routers, DHCP, and so on.

Reference implementations
Overview
Buying guide

OpenStack

- [Projects \(http://openstack.org/projects/\)](http://openstack.org/projects/)
- [OpenStack Security \(http://openstack.org/projects/openstack-security/\)](http://openstack.org/projects/openstack-security/)
- [Common Questions \(http://openstack.org/projects/openstack-faq/\)](http://openstack.org/projects/openstack-faq/)
- [Blog \(http://openstack.org/blog/\)](http://openstack.org/blog/)
- [News \(http://openstack.org/news/\)](http://openstack.org/news/)

Community

- [User Groups \(http://openstack.org/community/\)](http://openstack.org/community/)
- [Events \(http://openstack.org/community/events/\)](http://openstack.org/community/events/)
- [Jobs \(http://openstack.org/community/jobs/\)](http://openstack.org/community/jobs/)
- [Companies \(http://openstack.org/foundation/companies/\)](http://openstack.org/foundation/companies/)
- [Contribute \(http://docs.openstack.org/infra/manual/developers.html\)](http://docs.openstack.org/infra/manual/developers.html)

Documentation

- [OpenStack Manuals \(http://docs.openstack.org\)](http://docs.openstack.org)
- [Getting Started \(http://openstack.org/software/start/\)](http://openstack.org/software/start/)
- [API Documentation \(http://developer.openstack.org\)](http://developer.openstack.org)
- [Wiki \(https://wiki.openstack.org\)](https://wiki.openstack.org)

Branding & Legal

- [Logos & Guidelines \(http://openstack.org/brand/\)](http://openstack.org/brand/)
- [Trademark Policy \(http://openstack.org/brand/openstack-trademark-policy/\)](http://openstack.org/brand/openstack-trademark-policy/)
- [Privacy Policy \(http://openstack.org/privacy/\)](http://openstack.org/privacy/)
- [OpenStack CLA \(https://wiki.openstack.org/wiki/How_To_Contribute#Contributor_License_Agreement\)](https://wiki.openstack.org/wiki/How_To_Contribute#Contributor_License_Agreement)

Stay In Touch

(<https://twitter.com/OpenStack>) (<https://www.youtube.com/user/OpenStackFoundation>)

The OpenStack project is provided under the [Apache 2.0 license \(http://www.apache.org/licenses/LICENSE-2.0\)](http://www.apache.org/licenses/LICENSE-2.0). Openstack.org is powered by [Rackspace Cloud Computing \(http://rackspace.com\)](http://rackspace.com).