

VMware NSX-T Data Center Plugin for OpenStack Neutron Installation Guide

19 September 2019

VMware NSX-T Data Center 2.5



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2019 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

1	VMware NSX-T Data Center Plugin for OpenStack Neutron Installation Guide	4
2	Preparing to Install NSX-T Data Center Plugin for OpenStack	5
	Prerequisites	5
	System Requirements	5
	Neutron Plugin Comparision	6
3	Installing Neutron Basic Services With NSX-T Data Center Plug-In	9
	Install NSX-T Data Center Plugin for OpenStack on Ubuntu Systems	9
	Install NSX-T Data Center Plug-In for OpenStack on Red Hat Systems	10
4	Configuring OpenStack for NSX-T Data Center Plug-In	11
	Configure an OpenStack Neutron Network Node	11
	Edit the neutron.conf and nsx.ini Files	12
	Enable Client Certificate Based Authentication	13
	Enable DHCP and Metadata Proxy Services	13
	Sample Configuration File for NSX-T Data Center Plugin for OpenStack	15
	OpenStack Nova Controller Configuration	16
	OpenStack Nova Compute Configuration File	16
5	Configuring Neutron Advanced Services With NSX-T Data Center Plugin for OpenStack	18
	Configure OpenStack Octavia Load Balancer as a Service (LBaaS)	19
	Configure OpenStack Neutron's Load Balancer as a Service (LBaaS)	19
	Configure OpenStack Neutron's Firewall as a Service (FWaaS)	20
	Configure OpenStack Neutron's IPSec VPN as a Service (VPNaaS)	21
	Sample Configuration Files for Neutron Advanced Services	22
6	Appendix: NSX-T Data Center Plugin for OpenStack Configuration Properties	25

VMware NSX-T Data Center Plugin for OpenStack Neutron Installation Guide

1

This guide describes how to install and configure the NSX-T Data Center Plugin for OpenStack Neutron. The information includes step-by-step configuration instructions.

After it is configured and running, the VMware NSX-T Data Center Plugin plug-in allows OpenStack Neutron to realize and manage virtual networking resources in your NSX-T Data Center deployment. To do so successfully, you should be familiar with the NSX-T Data Center and OpenStack components and functions.

This guide provides information about Openstack Neutron plugins for NSX-T Policy and for NSX-T Manager. This is the first NSX-T Data Center release which provides an Openstack plugin for NSX-T Policy. For this release, the NSX-T Policy plugin can only be used for new installations.

To learn about these topics, see:

- *NSX-T Data Center Administration Guide*
- OpenStack documentation

Preparing to Install NSX-T Data Center Plugin for OpenStack

2

This chapter includes the following topics:

- [Prerequisites](#)
- [System Requirements](#)
- [Neutron Plugin Comparision](#)

Prerequisites

The support package provided by VMware for the OpenStack Neutron plugin only contains NSX-T specific artifacts. As a result, the OpenStack services of your choosing must be installed prior to attempting this installation process.

Follow the procedures in this document to install and configure the NSX-T Data Center Plugin for OpenStack Neutron to integrate with your NSX deployment. This procedure assumes that VMware NSX-T Data Center has been installed and configured on the NSX-T transport nodes.

Internet connectivity, or access to a local distribution repository mirror, is required during Neutron service installation to ensure the appropriate dependencies can be downloaded, installed and configured as part of the installation process.

System Requirements

The NSX-T Data Center Plugin for OpenStack support is implemented as a Neutron plugin. The VMware NSX plugin class used when configuring Neutron depends on the version of NSX you are using.

Supported Hypervisor versions for vSphere and KVM (Ubuntu, Red Hat Enterprise Linux, CentOS...) are listed in the *NSX-T Data Center Installation Guide*.

The NSX-T Data Center Plugin for OpenStack has the following specific requirements regarding compatible OpenStack software versions.

OpenStack Distribution for NSX-T Policy Plugin	Version
Open Source Edition	Stein

OpenStack Distribution for NSX-T Manager Plugin	Version
Open Source Edition	Rocky
Open Source Edition	Stein
Red Hat OpenStack Platform	Red Hat OpenStack Version 13 with the associated version of Red Hat Enterprise Linux.

Neutron Plugin Comparision

Starting with VMware NSX-T Datacenter 2.5, two plugins are available for integrating Openstack Neutron with NSX-T:

- The NSX-T Policy plugin interacts with NSX-T policy manager, using intent-based API abstractions. This is a new plugin and is the recommended choice for new installations.
- The NSX-T Manager plugin interacts with NSX-T manager, using imperative APIs. This is the existing NSX-T plugin and must be used for existing installations, as well as for use cases not yet covered by the NSX-T Policy plugin

Table 2-1. Plug-in Feature Comparison

Networking and Security Features	NSX-T MP Plugin	NSX-T Policy	Descirption
Switching			
Overlapping IP subnets support	Yes	Yes	Each project can dynamically create networks that are private to the project. These networks can have IP subnets that overlap with each other.
DHCP	Yes	Yes	Instances have automaic addressing via DHCP.
Static IPv6 address binding	No	Yes	
Routing			
Logical routing	Yes	Yes	Enable routing among multiple private logical networks, as well as between a logical network and an external network.
IPv6 logical routing	No	Yes	Enable routing among multiple private IPv6 logical networks, as well as between a logical network and an external network

Table 2-1. Plug-in Feature Comparison (continued)

Networking and Security Features	NSX-T MP Plugin	NSX-T Policy	Description
External networks	Yes	Yes	Networks that provide external access to the instances. Private networks will be uplinked to the external network via a router to provide external access to the instances on the private networks.
IPv6 external networks	Yes	Yes	External network with IPv6.
Static routes	Yes	Yes	Insert a static route.
IPv6 static routes	No	Yes	External network with IPv6.
Floating IP for instances	Yes	Yes	Assign public routable IP addresses to instances to enable external access in to the instances.
No-NAT router	Yes	Yes	No-NAT routing topology.
IPv6 no-NAT router	No	Yes	The No-NAT topology is the only routing topology supported by OpenStack with IPv6. NAT with IPv6 is not supported.
Neutron Router dual stack interfaces	No	Yes	Support of IPv4 and IPv6 dual stack on the same interfaces of a Neutron Router.
IPv6 SLAAC	No	Yes	Support of stateless address autoconfiguration.
Security			
Firewalling - security groups	Yes	Yes	OpenStack security groups (with NSX, security group are used + DFW rules created using those SG. This allow micro-segmentation)
IPv6 firewalling (security groups)	No	Yes	Neutron security group with IPv6.
Port security	Yes	Yes	Neutron Port Security is implemented using NSX SpoofGuard capabilities.

Table 2-1. Plug-in Feature Comparison (continued)

Networking and Security Features	NSX-T MP Plugin	NSX-T Policy	Description
IPv6 port security	No	Yes	Neutron Port Security is implemented using NSX SpoofGuard capabilities. This allows for allowed_address_pairs and an IPv6 subnet mapping to a port
Firewalling (L3 FWaaS)	Yes	Yes	
IPv6 Firewalling (L3 FWaaS)	No	Yes	
Other services			
Load balancing	Yes	Yes	
Quality of service	Yes	Yes	
DNS	Yes	Yes	
VPNaas	Yes	No	

Upgrades

There is no migration path from Openstack Neutron with NSX-T manager plugin to Openstack Neutron with NSX-T Policy plugin. When upgrading, existing installations should keep running the NSX-T Manager plugin. A migration path from NSX-T Manager to NSX-T Policy will become available in future releases. The NSX-T Policy plugin is the recommended solution for new installations as it includes unique features (IPv6); furthermore, moving forward new features will be available exclusively for the NSX-T plugin.

Installing Neutron Basic Services With NSX-T Data Center Plug-In

3

This chapter includes the following topics:

- [Install NSX-T Data Center Plugin for OpenStack on Ubuntu Systems](#)
- [Install NSX-T Data Center Plug-In for OpenStack on Red Hat Systems](#)

Install NSX-T Data Center Plugin for OpenStack on Ubuntu Systems

The NSX-T Data Center Plugins are distributed as debian (.deb) packages for Ubuntu based Linux distributions.

Prerequisites

The NSX-T Data Center Plugins are distributed as debian (.deb) packages for Ubuntu based Linux distributions.

The following instructions apply both to the NSX-T Manager and NSX-T Policy plugins.

- If the installation of the debian package fails due to dependency errors, it may be necessary to install the python-tooz and python-oslo.vmware packages. Tooz is a Python library that provides abstractions for distributed coordination primitives. Its primary goal is to handle groups and membership of these groups in distributed systems. The Oslo VMWare library provides support for common VMWare operations and APIs. For example: `sudo apt-get install python-oslo.vmware`.

Procedure

- 1 Download the .deb files: the NSX Neutron plugins and the NSX Neutron common library.
- 2 Copy the files to the Neutron network node.
- 3 Install the package using the dpkg command in the same directory as the .deb file.

Version numbers in the following example may differ depending on the release that is selected upon download:

- `sudo dpkg -i python-vmware-nsxlib_12.0.0.9797177-1_all.deb`
- `sudo dpkg -i openstack-vmware-nsx_12.0.0.9797177-1_all.deb`

- 4 Install the Firewall as a Service (FWaaS) package. This does not need to be enabled after installation.

Version numbers in the following example may differ depending on the release that is selected upon download:

- `sudo apt-get install python-neutron-fwaas`

Install NSX-T Data Center Plug-In for OpenStack on Red Hat Systems

The Install NSX-T Data Center Plug-Ins for OpenStack are packaged as .rpm files for Red Hat based Linux distributions.

The .rpm packages can be found on the NSX-T Data Center download page under **Drivers and Tools**. These instructions are for installations that do not leverage TripleO, otherwise refer to the dedicated guide for Red Hat OpenStack.

Procedure

- 1 Download the .rpm files: the NSX Neutron plugins and the NSX Neutron common library.
- 2 Copy it to the Neutron network node you on which you wish to install the plugin.
- 3 Install the package using the rpm command in the same directory as the .rpm file.

Version numbers in the following example may differ depending on the release that is selected upon download:

- `sudo rpm -i python-vmware-nsxlib_12.0.0.9797177-1_all.rpm`
- `sudo rpm -i vmware-nsx-12.0.0.9797177-1.noarch.rpm`

- 4 Install the Firewall as a Service (FWaaS) package. This does not need to be enabled after installation.

- `sudo yum install python-neutron-fwaas`

Configuring OpenStack for NSX-T Data Center Plug-In

4

- [Configure an OpenStack Neutron Network Node](#)

The configuration described in this section describes the Neutron network node configuration.

- [Sample Configuration File for NSX-T Data Center Plugin for OpenStack](#)

Configuration files are usually located at `/etc/neutron/plugins/vmware/nsx.ini`.

- [OpenStack Nova Controller Configuration](#)

The configuration described in this section supplements Nova Controller nodes configuration with information about NSX-T Data Center.

- [OpenStack Nova Compute Configuration File](#)

To edit Nova configuration files use NSX-T.

Configure an OpenStack Neutron Network Node

The configuration described in this section describes the Neutron network node configuration.

The documentation refers to the NSX-T Policy plugin. However, this section will also highlight settings specific to the NSX-T Manager plugin.

Two default configuration file paths are relevant:

- `/etc/neutron/neutron.conf` -- Neutron's configuration file.
- `/etc/neutron/plugin/vmware/nsx.ini` -- VMware NSX Neutron plugin's configuration file.

- [Edit the neutron.conf and nsx.ini Files](#)

These files must be edited with information related to the NSX-T environment to enable the Neutron plugin to interact with the NSX-T deployment.

- [Enable Client Certificate Based Authentication](#)

Neutron client certificate based authentication to NSX Manager is supported.

- [Enable DHCP and Metadata Proxy Services](#)

With the NSX-T Data Center plugin, the OpenStack reference DHCP implementation is replaced by the NSX-T Data Center native DHCP server. The NSX-T Data Center platform also provides a proxy server for accessing Nova metadata.

Edit the neutron.conf and nsx.ini Files

These files must be edited with information related to the NSX-T environment to enable the Neutron plugin to interact with the NSX-T deployment.

Procedure

- 1 Edit the `neutron.conf` file to set the core Neutron plugin

[DEFAULT] core_plugin = vmware_nsxp. To enable the NSX-T Manager plugin:

[DEFAULT] core_plugin = vmware_nsxv3

- 2 Edit the `nsx.ini` configuration file to configure the plugin for your NSX deployment.

NSX-T OpenStack Plugin properties go under the `[nsx_p]` section of the `nsx.ini` configuration file.

The configuration properties below also apply to the NSX-T Manager plugin, and are specified in the `[nsx_v3]` configuration section.

The minimal set of configuration properties that need to be defined are:

Variable	Description
<code>nsx_api_managers</code>	This parameter allows a list of comma separated manager endpoints.
<code>nsx_api_user</code>	Administrator NSX-T Manager user name, usually admin.
<code>nsx_api_password</code>	Administrator NSX-T Manager password.
<code>insecure</code>	Set to False to enforce verification of NSX Manager server certificate. Defaults to True.
<code>ca_file</code>	CA bundle files to use in verifying the NSX Manager server certificate. This option is ignored if "insecure" is set to True. If "insecure" is set to False and this option is unset, the system root CAs will be used to verify the server certificate.
<code>nsx_api_managers</code>	The name or UUID of the default NSX overlay transport zone that will be used for creating Neutron networks. It needs to be created in NSX before starting Neutron.
<code>default_tier0_router</code>	This must be a Policy Manager NSX-T Tier0 name gateway router or UUID to which future OpenStack Logical Routers (NSX-T Tier1) will be connected (under "Routing/Routers").
<code>dhcp_profile</code>	Enter either a UUID or name. See Create a DHCP Profile in NSX Manager .
<code>metadata_proxy</code>	Enter either a UUID or name. See Create a Metadata Proxy .

- 3 Restart Neutron to pick-up the changes in the `nsx.ini` file by running the command: `ps-aux | grep neutron`.

- 4 Verify that `nsx.ini` and `neutron.conf` are present in the output. Note that Neutron accepts one or more configuration files in the command line. These files are merged when the configuration is parsed so that the configuration file structure can reflect a particular user's preferences.

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

Enable Client Certificate Based Authentication

Neutron client certificate based authentication to NSX Manager is supported.

Client certificate-based authentication enables the Neutron plugin to login as the principal identity with the Enterprise Administrator role. Other principal identities are not able to edit resources created by the neutron principal identity, thus protecting from accidental errors such as the deletion of a logical router associated to a neutron router. See "View Principal Identity" in NSX-T for Data Center Administration Guide for more information.

Procedure

- 1 To enable client certificate authentication, define the following in the `nsx.ini` file:

- `nsx_use_client_auth = True`
- `nsx_client_cert_storage = nsx-db`
- `nsx_client_cert_file = <file to store certificate and private key>`

- 2 Restart Neutron to pick-up the changes in the `nsx.ini` file, by running the command: `service neutron-server restart`.

Verify that the Neutron Server is using both the `neutron.conf` and `nsx.ini` files by running the following command:

- `ps -aux |grep neutron`

Verify that `nsx.ini` and `neutron.conf` are present in the output. For example:

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

Enable DHCP and Metadata Proxy Services

With the NSX-T Data Center plugin, the OpenStack reference DHCP implementation is replaced by the NSX-T Data Center native DHCP server. The NSX-T Data Center platform also provides a proxy server for accessing Nova metadata.

These operations must be performed, regardless of the NSX-T Plugin configured in the previous step.

- [Create a DHCP Profile in NSX Manager](#)

A DHCP server profile specifies an NSX Edge cluster or members of an NSX Edge cluster. A DHCP server with this profile services DHCP requests from VMs on logical switches that are connected to the NSX Edge nodes that are specified in the profile.

- [Create a Metadata Proxy](#)

With a metadata proxy server, VM instances can retrieve instance-specific metadata from an OpenStack Nova API server.

- [Edit the `nsx.ini` File](#)

The new variables `native_dhcp_metadata`, `metadata_proxy`, and `dhcp_profile` need to be specified in `nsx.ini` consuming those profiles.

Create a DHCP Profile in NSX Manager

A DHCP server profile specifies an NSX Edge cluster or members of an NSX Edge cluster. A DHCP server with this profile services DHCP requests from VMs on logical switches that are connected to the NSX Edge nodes that are specified in the profile.

Prerequisites

To enable NSX-T Data Center's native DHCP Server, a DHCP profile must be created in NSX-T Data Center and passed to the Neutron plugin configuration in `nsx.ini`. Ensure the neutron dhcp service (q-dhcp in devstack), and the metadata agent (q-meta in devstack) are not running. In the `neutron.conf` file, set to `False`.

Procedure

- 1 From your browser, log in with admin privileges to an NSX Manager at `https://nsx-manager-ip-address`.
- 2 Select **Advanced Networking & Security > DHCP** from the navigation panel.
- 3 Select **Server Profiles** and click **Add**.
- 4 Enter a name and optional description.
- 5 Select an **Edge Cluster** from the drop-down menu.
- 6 Click **Add**.

Create a Metadata Proxy

With a metadata proxy server, VM instances can retrieve instance-specific metadata from an OpenStack Nova API server.

The NSX platform provides a proxy server for accessing Nova metadata. The proxy will capture all requests made at the 169.254.269.254 address, and forward them to the nova metadata server endpoint specified in the NSX-T metadata proxy configuration

Prerequisites

Edge Node used for the metadata proxy must have IP connectivity to the metadata server from their management IP addresses.

Procedure

- 1 From your browser, log in with administrator privileges to an NSX Manager at <https://nsx-manager-ip-address>.
- 2 Select **Advanced Networking > DHCP** from the navigation panel.
- 3 Select **Metadata Proxies** and click **Add**.
- 4 Enter a **Name** and optional description.
- 5 Enter the **Nova Server URL** as `http://<openstack_controller>:8775`. If the Metadata Proxy Server is listening on a port other than the default 8775, update the URL with the correct port. Ports can be found on the controller node in the Nova API configuration file `/etc/nova.conf`, looking up the `metadata_listen_port` parameter. If the configuration needs to be changed, restart `n-api` or `nova` server.
- 6 Enter the **Secret** parameter.
- 7 Select an **Edge Cluster** from the drop-down menu.
- 8 Click **Add**.

Edit the nsx.ini File

The new variables `native_dhcp_metadata`, `metadata_proxy`, and `dhcp_profile` need to be specified in `nsx.ini` consuming those profiles.

- `dhcp_profile = <UUID or name - DHCP>`
- `native_dhcp_metadata = True`
- `metadata_proxy = <UUID or name - MetaData Proxy>`
- `native_metadata_route = 169.254.169.254/31`

Sample Configuration File for NSX-T Data Center Plugin for OpenStack

Configuration files are usually located at `/etc/neutron/plugins/vmware/nsx.ini`.

Below is a sample configuration file:

```
[nsx_p]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
```

```
default_tier0_router = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
default_overlay_tz_ = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
default_vlan_tz = f74b5dab-dad3-47d2-b46e-57a1eeb5fde3
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

```
[nsx_v3]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
default_tier0_router_uuid = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
default_overlay_tz_uuid = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

OpenStack Nova Controller Configuration

The configuration described in this section supplements Nova Controller nodes configuration with information about NSX-T Data Center.

Below is a Nova Controller sample configuration file which is usually located on `/etc/nova/nova.conf` on control node.

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
metadata_proxy_shared_secret = VMware1!
service_metadata_proxy = True
```

OpenStack Nova Compute Configuration File

To edit Nova configuration files use NSX-T.

The configuration described in this section supplements Nova Compute nodes configuration with information about NSX-T Data Center.

Below is a Nova Compute sample configuration file, which is usually located on `/etc/nova/nova.conf` on compute nodes.

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
#for KVM
ovs_bridge = nsx-managed
```

Configuring Neutron Advanced Services With NSX-T Data Center Plugin for OpenStack

5

OpenStack Neutron's Load Balancer as a Service (LBaaS), Firewall as a service (FWaaS), and IPsec VPN as a Service (VPNaaS) are also referred to as Neutron advanced services.

Starting with NSX-T Data Center 2.5 NSX-T plugins (both Policy and Manager) support the Octavia load balancing service, which replaces the deprecated Neutron LBaaS service.

The following information assumes that you have NSX-T Data Center 2.4, and OpenStack Stein 14.0.

The current versions supported are listed in [System Requirements](#)

To enable Neutron advanced services NSX-T Data Center the following should be specified in the Neutron configuration:

- Enable service plugin for the specific service
- Configure service providers for the service
- Provide NSX-T Data Center driver, where applicable, and configuration entries specific to NSX-T Data Center

These options are specified in neutron configuration files, usually found in `/etc/neutron` (note: plugin-specific configuration files are usually found in `/etc/neutron/plugins/vmware`. Neutron accepts one or more configuration files in the command line. These files are merged when the configuration is parsed so that the configuration file structure can reflect a particular user's preferences. The following structure is usually employed:

- `neutron.conf` - core neutron options, basic configuration parameters (e.g.: API manager endpoint, transport zone identifiers), service plugin list.
- `neutron_lbass.conf` - load balancing service provider and options.
- `Octavia.conf` - load balancing provider and MQ topic. Only for Octavia Load Balancer service. This file is usually located in `/etc/Octavia/Octavia.conf` and is loaded by the Octavia service.
- `neutron_fwass.conf` - firewall service provider, driver, and driver options.
- `neutron_vpnaas.conf` - VPN service provider and other options.

Service providers are specified using the `service_provider` option. This option can be repeated multiple times in a neutron configuration for different kind of services, but there cannot be more than one default service provider for a given service type.

Service_provider configuration attribute structure: <SERVICE_TYPE>:<PROVIDER_CLASS>: [<DEFAULT>].

- [Configure OpenStack Octavia Load Balancer as a Service \(LBaaS\)](#)

These instructions are for configuring OpenStack consumption of NSX-T Data Center Load Balancer with Octavia.

- [Configure OpenStack Neutron's Load Balancer as a Service \(LBaaS\)](#)

- [Configure OpenStack Neutron's Firewall as a Service \(FWaaS\)](#)

These instructions are for configuring FWaaS v2.

- [Configure OpenStack Neutron's IPsec VPN as a Service \(VPNaaS\)](#)

This service enables OpenStack users to make Neutron networks accessible to remote site over secure VPN tunnels.

- [Sample Configuration Files for Neutron Advanced Services](#)

Configure OpenStack Octavia Load Balancer as a Service (LBaaS)

These instructions are for configuring OpenStack consumption of NSX-T Data Center Load Balancer with Octavia.

The same instructions apply both to the NSX-T Policy and NSX-T Manager plugin.

For general information about the Octavia load balancer service please refer to the official documentation.

Procedure

- 1 In `/etc/neutron/neutron.conf`, ensure the load balancing service plugin for NSX-T Data Center is not configured, ie: `vmware_nsx_lbaasv2` should not be present in the `service_plugins` list. If any changes are done to the `neutron.conf` file, restart the neutron service
- 2 In the `/etc/octavia/octavia.conf` specify the following:
 - a In the `[api_settings]` configuration setting, specify the NSX-T provider:
`default_provider_driver = vmwareedge` `enabled_provider_drivers = vmwareedge:NSX`
 - b In the `[oslo_messaging]` section, specify a topic for communication between Octavia and the NSX-T driver, which runs in the neutron process space.
- 3 Restart the Octavia API service.

Configure OpenStack Neutron's Load Balancer as a Service (LBaaS)

These instructions are for configuring OpenStack consumption of NSX-T Data Center Load Balancer with LBaaSv2. The same instructions apply both to the NSX-T Policy and NSX-T Manager plugin.

Procedure

- 1 Edit `/etc/neutron/neutron.conf` to add the load balancing service plugin for NSX-T Data Center, in the default configuration section: `service_plugins = vmware_nsx_lbaasv2, [...]`

Note that `service_plugins` is a list option. It is possible to specify multiple service plugins by separating their full class names, or shortcuts, with a comma.
- 2 Edit the `/etc/neutron/neutron-lbaas.conf` file with the following:
 - a Set the load balancer service driver for NSX-T Data Center, by setting the `service_provider` option in the `service_providers` configuration section: `service_plugins = LOADBALANCERV2:VMWareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default`
 - b Configure Keystone authentication parameters, if not already configured. These are used by the `neutron-lbaas` service and are not related to NSX-T Data Center integration. Note that the location of the Keystone endpoint should be provided.

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

- 3 Ensure the file `/etc/neutron/neutron-lbaas.conf` is added to the neutron server command line. This could be verified by running the command `ps -aux | grep neutron`, and verifying that `/etc/neutron/neutron-lbaas.conf` is present in the output.

If the file is not included, the neutron service launcher should be edited. Location and structure of service launchers depend on the particular OpenStack distribution used.

- 4 Restart the neutron service. The exact service name depends on the OpenStack distribution used.

Configure OpenStack Neutron's Firewall as a Service (FWaaS)

These instructions are for configuring FWaaS v2.

Procedure

- 1 Edit `/etc/neutron/neutron.conf` to add the firewall as a service plugin for NSX-T Data Center, in the default configuration section: `service_plugins = firewall_v2`

`Service_plugins` is a list option. Multiple service plugins can be specified by separating their full class names, or shortcuts, with a comma.

2 Edit the `/etc/neutron/neutron-fwaas.conf` file with the following:

- a Set the firewall as a service driver for NSX-T Data Center, by setting the `service_provider` option in the `service_providers` configuration section. `service_provider = FIREWALL_V2:fwaas_db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver:default`

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

The value of this option has a particular structure:

`<service_type>:<service_name>:<driver_class>:[<default>]`. `service_provider` is a “multi-string” option. Every time it is specified, the value of the option is added to a list. Multiple service providers can be specified by setting the `service_provider` option for each of them.

- b Switch on firewall as a service, by setting `enabled = True` in the FWaaS configuration section.
 - c Set the FWaaS device driver for NSX-T Data Center, by setting: `driver = vmware_nsxp_edge_v2` in the FWaaS configuration section.
 - d Set the FWaaS device driver for the NSX Manager plugin by setting `driver = vmware_nsxv3_edge_v2` in the FWaaS configuration section.
- 3** Check that the file `/etc/neutron/neutron-fwaas.conf` is added to the neutron server command line. This can be verified by running `ps -aux | grep neutron` and verifying that `/etc/neutron/neutron-fwaas.conf` is present in the output.

If the file is not included, the neutron service launcher should be edited. The location and structure of service launchers depend on the particular OpenStack distribution used.

- 4** Restart the neutron service. The specific service name depend on the OpenStack distribution used.

Configure OpenStack Neutron’s IPsec VPN as a Service (VPNaaS)

This service enables OpenStack users to make Neutron networks accessible to remote site over secure VPN tunnels.

The VPNaaS driver is not available for the NSX-T Policy plugin. The following instructions only apply to the NSX-T manager plugin.

Procedure

- 1 Edit `/etc/neutron/neutron.conf` to add the IPsec VPN as a service plugin for NSX-T Data Center, in the default configuration section: `service_plugins = vmware_nsx_vpnaas, [...]`

`service_plugins` is a list option. It is possible to specify multiple service plugins by separating their full class names, or shortcuts, with a comma.
- 2 Edit the `/etc/neutron/neutron-vpnaas.conf` file with the following: Set the load VPNservice driver for NSX-T Data Center, by setting the `service_provider` option in the `service_providers` configuration section. `service_provider = VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default`

The value of this option has a particular structure:
`<service_type>:<service_name>:<driver_class>:[<default>]`. `service_provider` is a “multi-string” option. Every time it is specified, the value of the option is added to a list. It is possible to specify multiple service providers by setting the `service_provider` option for each of them.
- 3 Ensure the file `/etc/neutron/neutron-vpnaas.conf` is added to the neutron server command line. This can be verified by running `ps -aux | grep neutron` and verifying that `/etc/neutron/neutron-vpnaas.conf` is present in the output.

If the file is not included, the neutron service launcher should be edited. The location and structure of service launchers depend on the particular Openstack distribution used.
- 4 Restart the neutron service. The specific service name depends on the OpenStack distribution used.

Sample Configuration Files for Neutron Advanced Services

```
[DEFAULT]
ovs_integration_bridge = nsxvswitch
dhcp_agent_notification = False
notify_nova_on_port_data_changes = True
notify_nova_on_port_status_changes = True
core_plugin = vmware_nsxv3
service_plugins =
vmware_nsx_lbaasv2,vmware_nsx_vpnaas,neutron_fwaas.services.firewall.fwaas_plugin_v2.FirewallPluginV2
[...]
neutron_vpnaas.conf
[DEFAULT]
[service_providers]
service_provider =
VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default
neutron_fwaas.conf
[DEFAULT]
[quotas]
# Number of firewalls allowed per tenant. A negative value means unlimited.
# (integer value)
#quota_firewall = 10
```

```

# Number of firewall policies allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_firewall_policy = 10
# Number of firewall rules allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_firewall_rule = 100

[service_providers]
service_provider =
FIREWALL_V2:fwaas-db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver
:default
[fwaas]
enabled = True
driver = vmware_nsxv3_edge_v2
neutron_lbaas.conf
[DEFAULT]
[quotas]
# Number of LoadBalancers allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_loadbalancer = 10

# Number of Loadbalancer Listeners allowed per tenant. A negative
# value means unlimited. (integer value)
#quota_listener = -1
# Number of pools allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_pool = 10
# Number of pool members allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_member = -1
# Number of health monitors allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_healthmonitor = -1
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_ip>/identity/v3
[service_providers]
service_provider =
LOADBALANCER_V2:VMwareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default

```

```

Octavia.conf
[DEFAULT]
verbose = True
transport_url = rabbit://<amqp_user>:<amqp_password>@<amqp_node>:5672/
debug = True

[api_settings]
default_provider_driver = vmwareedge
enabled_provider_drivers = vmwareedge:NSX
bind_port = 9875

```

```

api_handler = queue_producer
bind_host = 0.0.0.0

[database]
connection = mysql+pymysql://root:<db_password?@<db_node>:3306/octavia

[keystone_auth_token]
signing_dir =
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = service
user_domain_name = Default
password = <password>
username = octavia
auth_url = http://<keystone_node>/identity
auth_type = password

[certificates]
server_certs_key_passphrase = insecure-key-do-not-use-this-key
ca_private_key_passphrase = foobar
ca_private_key = /etc/octavia/certs/private/cakey.pem
ca_certificate = /etc/octavia/certs/ca_01.pem

[controller_worker]
amp_ssh_key_name = octavia_ssh_key
amp_image_tag = amphora
network_driver = allowed_address_pairs_driver
compute_driver = compute_nova_driver
amphora_driver = amphora_haproxy_rest_driver
workers = 2
amp_active_retries = 100
amp_active_wait_sec = 2

[oslo_messaging]
topic = vmwarensxv_edge_lb
rpc_thread_pool_size = 2

[house_keeping]
load_balancer_expiry_age = 3600

[service_auth]
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = admin
user_domain_name = Default
password = openstack
username = admin

```


Appendix: NSX-T Data Center Plugin for OpenStack Configuration Properties

6

Table 6-1. Configuration Properties

Section	Variable	Description
nsx_p	nsx_api_managers	The IP address of one or more NSX Managers separated by commas. The IP address should be in the following form: [<scheme>://<ip_address>[:<port>]. If scheme is not provided https is used. If a port is not provided, port 80 is used for http and port 443 for https.
	nsx_use_client_auth	Boolean. Set to True to enable client certificate authentication
	nsx_client_cert_file	Path to a file containing client certificate and private key, in PEM format.
	nsx_client_cert_pk_password	(Optional) Password for decrypting the private key.
	nsx_api_user	The username used to access the for NSX Manager API.
	nsx_api_password	The password used to access the NSX Manager API.
	dns_domain	Domain to use for building the hostnames.
	default_overlay_tz	default_edge_cluster
	default_vlan_tz	(Optional) Only required when creating VLAN or flat provider networks. The UUID or name of the default NSX VLAN transport zone that is used for bridging between Neutron networks if no physical network has been specified.
	edge_cluster	(Optional) Specifying an edge cluster for Tier1 routers to which to connect, other that the edge cluster it is connected to.
	retries	(Optional) The maximum number of times to retry API requests upon stale revision errors.

Table 6-1. Configuration Properties (continued)

Section	Variable	Description
	ca_file	(Optional) Specify a CA bundle file to use in verifying the NSX Manager server certificate. This option is ignored if "insecure" is set to True. If "insecure" is set to False and ca_file is unset, the system root CAs will be used to verify the server certificate.
	insecure	(Optional) If true, the NSX Manager server certificate is not verified. If false the CA bundle specified via "ca_file" will be used or if unset the default system root CAs will be used.
	http_timeout	(Optional) The time in seconds before aborting a HTTP connection to a NSX Manager.
	http_read_timeout	(Optional) The time in seconds before aborting a HTTP read response from a NSX Manager.
	http_retries	(Optional) Maximum number of times to retry a HTTP connection.
	concurrent_connections	(Optional) Maximum number of connection connections to each NSX Manager.
	conn_idle_timeout	(Optional) The amount of time in seconds to wait before ensuring connectivity to the NSX manager if no Manager connection has been used.
	default_tier0_router	(Optional) The UUID or name of the default Tier 0 router that is used for connecting to Tier 1 logical routers and configuring external networks.
	metadata_on_demand	(Optional) If True, an internal metadata network is created for a router only when the router is attached to a DHCP-disabled subnet.
	dhcp_profile	(Optional) The UUID of the NSX DHCP Profile that is used to enable native DHCP service. It needs to be created in NSX before starting Neutron with the NSX plugin.
	locking_coordinator_url	(Optional) URL for distributed locking coordination resource for lock manager. This value is passed as a parameter to tooz coordinator. By default, the value is None and oslo_concurrency is used for single-node lock management.

Table 6-1. Configuration Properties (continued)

Section	Variable	Description
	realization_max_attempts	(Optional) Maximum number retry attempts allowed while waiting for a resource to be realized. Default: 50
	realization_wait_sec	(Optional) Number of seconds between attempts for a resource to be realized. Default: 1 second