VMware NSX OpenStack Plugin

Installation and Configuration

NSX-T 2.2

Table of Contents

Overview

Related Documentation

Prerequisites

System Requirements

Installing the NSX Plugins

- Configuration File Layout

Configuring the NSX-T OpenStack Plugin

- Configuring the Layer 2 Gateway with the NSX-T OpenStack Plugin
- L2 Gateway Configuration
- DHCP and MetaData Proxy Services
  - Create a DHCP Profile
  - Create a MetaData Proxy
  - Steps to Deploy OpenStack
- Configuring the Neutron DHCP Agent
- NSX-T Plugin Configuration Properties
- Sample nsx.ini
  - NSX-T:

Configuring NSX-T with Red Hat OpenStack Platform

Overview

This document outlines the VMware NSX OpenStack Neutron plugin installation and configuration process. Once configured and running, the NSX plugin will allow OpenStack Neutron to realize and manage virtual networking resources in your NSX deployment.

Related Documentation

This document assumes the reader is already familiar with NSX and OpenStack concepts.

To learn about these topics, please see:

- VMware NSX Installation, Administration, and API Guides: https://www.vmware.com/support/pubs/nsxt_pubs.html
- NSX-T: https://docs.vmware.com/en/VMware-NSX-T/index.html
- OpenStack documentation: http://docs.openstack.org/
- OpenStack networking (Neutron) documentation: http://docs.openstack.org/admin-guide/index.html
- VMware NSX plugin REST API reference: https://github.com/openstack/vmware-nsx/blob/master/api-ref/rest.md

Prerequisites

The support package provided by VMware for the OpenStack Neutron plugins only contains NSX plugin 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 VMware NSX plugin to integrate with your NSX deployment.

Note that 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 VMware NSX OpenStack Neutron support is implemented as a Neutron plugin; one plugin per supported version of NSX. The VMware NSX plugin class used when configuring Neutron depends on the version of NSX you are using.

The VMware NSX OpenStack Neutron plugin has the following specific requirements regarding compatible OpenStack software versions.

OpenStack	Version
Open Source Edition	Pike 11
Open Source Edition	Queens 12
Red Hat OpenStack Platform	Version 10 with the associated version of Red Hat Enterprise Linux running Red Hat OpenStack Platform components.
Canonical OpenStack	16.04 LTS based

Supported Hypervisor versions for Ubuntu and Red Hat Enterprise Linux are listed in the *NSX Installation Guide*.

Installing the NSX Plugins

The NSX plugin for OpenStack Neutron is packaged as '.deb' for '.rpm' files for Ubuntu and Red Hat based Linux distributions and delivered from the Drivers and Tools area of the NSX downloads area on vmware.com.

Before installing the debian packages, python-tooz and python-oslo.vmware must be installed. Tooz is a Python library that provides a coordination API. 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-tooz
> sudo apt-get install python-oslo.vmware
```

To install the NSX Neutron .deb packages on an Ubuntu based OpenStack system, download the `.deb` file and copy it to the Neutron network node you on which you wish to install the plugin. Then install the package using the dpkg command in the same directory as the .deb file.

For example:

```
> sudo dpkg -i openstack-vmware-nsx_10.0.1.6517389-1_all.deb
```

Note that the version numbers may differ depending on the release that is selected upon download.

To install the NSX Neutron .rpm packages on an Red Hat based OpenStack system, download the `.rpm` file and copy it to the Neutron network node you on which you wish to install the plugin. Then install the package using the rpm command in the same directory as the .rpm file.
For example:

```
> sudo rpm -i openstack-vmware-nsx_9.1.1.4789047-1_all.rpm
```

Note that the version numbers may differ depending on the release that is selected upon download.

Now that the plugin files are installed, edit the file (as root using `sudo`) `/etc/default/neutron-server` and set the `NEUTRON_PLUGIN_CONFIG` variable to the NSX plugin configuration file (see the Configuration File Layout below).

Note: Starting with Pike, you must install the Firewall as a Service (FWaaS) package. This does not need to be enabled after installation.

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

Configuration File Layout

The default configuration file paths that are relevant:

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

Configuring the NSX-T OpenStack Plugin

The configuration documentation herein is intended to supplement your OpenStack distributions configuration documentation. In particular, the configuration described in this section supplements Neutron network node configuration.

To enable the NSX-T OpenStack Plugin, edit the `neutron.conf` file (see Configuration File Layout). To set the core plugin Neutron use:

```
[DEFAULT]
core_plugin = vmware_nsx.plugin.NsxV3Plugin
```

Now that Neutron has been configured to use the NSX-T OpenStack Plugin, edit the `nsx.ini` configuration file (see Configuration File Layout) to configure the plugin for your NSX deployment. NSX-T OpenStack Plugin properties go under the `[nsxv3]` section of `nsx.ini`.

The minimal set of configuration properties you need to define are listed below:

- `nsx_api_managers`: IP address of NSX-T Manager
- `nsx_api_user`: Administrator NSX-T Manager user name, usually admin
- `nsx_api_password`: Administrator NSX-T Manager password
- `default_overlay_tz`: Administrator NSX-T Manager user name, usually admin
- `default_tier0_router`: NSX-T Tier0 UUID to which future OpenStack Logical Routers (NSX-T Tier1) will be connected (under "Routing/Routers")
- `default_edge_cluster`: NSX-T Edge Cluster UUID where future OpenStack Logical Routers (NSX-T Tier1) will be deployed (under "Fabric/Nodes/Edge Clusters")
- `dhcp_profile`: See explanations below.
- `native_dhcp_metadata = True`: See explanations below.
- `metadata_proxy`: See explanations below.

Client certificate authentication to NSX is supported. Client certification is disabled by default.

To enable client 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>
```

Other properties may be set depending on your environment and desired topology.

Once you've configured `nsx.ini`, restart Neutron to pick-up the changes:

```
service neutron-server restart
```

Configuring the Layer 2 Gateway with the NSX-T OpenStack Plugin

L2 gateways bridge two or more L2 networks, and make them appear as a single L2 broadcast domain. In OpenStack Neutron, the L2 gateway constructs help in extending the tenants logical overlay (GENEVE) network into VLAN networks, which may or may not be managed by OpenStack.

Terminology

1. **Bridge Cluster (BC)**: A collection of transport nodes that will perform the bridging for overlay network to VLAN networks. A bridge cluster may have one or more transport nodes.
2. **Bridge Endpoint (BE)**: Identifies the physical attributes of a bridge. It is analogous to a VIF i.e it can be used as an attachment to a logical port. A BE will usually consists of following tuple : <BC-id, vlan-id, …> where BC-id is the bridge cluster id. Additional properties may be present. When a logical port is attached to a bridge endpoint, logically it will create a bridge that will bridge the VNI on a logical port to the VLAN on a BE.

L2 Gateway Configuration

1. Create a bridge cluster and add transport nodes to it in the NSX Manager. This returns a BC-UUID to the administrator, which should be configured in the nsx.ini file <default_bridge_cluster> = <bc-uuid>
2. The administrator then creates a bridge endpoint on a bridge cluster, and passes the VLAN ID, which will be connected to the gateway.
3. Create a logical port on the overlay network with the attachment_type: BRIDGEENDPOINT and attach it to the bridge endpoint created in step 2. The bridge-endpoint uuid present in the logical port attachment is used to find the bridged VLAN.

DHCP and MetaData Proxy Services

In NSX 1.1.0 and later, users can select the native NSX DHCP and Metadata services supported by NSX backend. Comparing to Neutron DHCP/Metadata services, the native DHCP/Metadata services do not need to create namespace and start metadata proxy for each Neutron network. Therefore it reduces the overheads on the controller nodes.

To enable native NSX DHCP and Metadata services the administrator, users needs to disable `q-dhcp` and `q-meta` services, and set `dhcp_agent_notification` to `False` in `Neutron.conf`. In addition, there new variables `native_dhcp_metadata`, `metadata_proxy`, and `dhcp_profile` need to be specified in `nsx.ini` (as described above).

Create a DHCP Profile

1. Log in to NSX Manager.
2. Click DHCP.
3. Click Server Profiles.
4. Click Add.
5. Enter Name, Edge Cluster (select from the list), and Members (optional, select from the list).
6. Click Save.

Create a MetaData Proxy

1. Log in to NSX Manager.
2. Click DHCP.
3. Click MetaData Proxies.
4. Click Add.
5. Enter Name, Edge Cluster (select from the list), and Members (optional, select from the list).
6. Enter Nova Server URL as http://<openstack_controler>:8775 (if using a port number other than 8775, add "metadata_listen_port = <new_port_number>" in /etc/nova.conf and restart n-api or nova server).
7. If prompted, enter secret.
8. Click Save.

Configuring the Neutron DHCP Agent

The NSX-T OpenStack Plugin supports the native NSX DHCP service. If you choose not to use native NSX DHCP support, these are the steps required to configure the OpenStack DHCP agent.

To configure the DHCP agent, edit the `dhcp_agent.ini` file located in the same directory as `neutron.conf` (see Configuration File Layout). Once the configuration changes are made, the DHCP agent needs to be restarted by restarting the neutron agent.

```
[DEFAULT]
ovs_integration_bridge = <nsx-managed>
enable_metadata_network = True
enable_isolated_metadata = True
interface_driver =
neutron.agent.linux.interface.OVSInterfaceDriver
ovs_use_veth = True
```

NSX-T Plugin Configuration Properties

Section	Variable	Description
nsx_v3	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_adress>[:<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.
	dns_domain	Domain to use for building the hostnames.
	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.
	`default_overlay_tz`	The UUID or name of the default NSX overlay transport zone that is used for creating tunneled or isolated Neutron networks. If no physical network is specified when creating a logical network, this transport zone will be used by default.
	`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.
	`default_edge_cluster`	(Optional) Default Edge Cluster UUID or name.
	`retries`	(Optional) The maximum number of times to retry API requests upon stale revision errors.
	`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 tier0 router that is used for connecting to tier1 logical routers and configuring external networks.
	default_bridge_cluster	(Optional) The UUID or name of the default NSX bridge cluster that is used to perform L2 gateway bridging between VXLAN and VLAN networks. If the default bridge cluster UUID is not specified, the administrator has to manually create a L2 gateway corresponding to an NSX Bridge Cluster using L2 gateway APIs. This field must be specified on one of the active Neutron servers only.
	number_of_nested_groups	(Optional) The number of nested groups which are used by the plugin. Each Neutron security-groups is added to one nested group, and each nested group can contain a maximum of 500 security-groups, therefore, the maximum number of security groups that can be created is 500 * number_of_nested_groups. The default

		is 8 nested groups, which allows a maximum of 4k security-groups. To allow the creation of more security-groups, modify this figure.
	`metadata_mode`	(Optional) Acceptable values are: **access_network**: enables a dedicated connection to the metadata proxy for metadata server access via Neutron router. **dhcp_host_route**: enables host route injection via the dhcp agent. This option is only useful if running on a host that does not support namespaces otherwise access_network should be used.
	`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.
	`native_dhcp_metadata`	(Optional) If true, DHCP and metadata proxy services will be provided by NSX.
	`metadata_proxy`	(Optional) The UUID of the NSX Metadata Proxy that is used to enable native metadata service. It needs to be created in NSX before starting Neutron with the NSX plugin.
	`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.
`Default`	`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.

Sample nsx.ini

Below is a sample configuration file, which is usually located at `/etc/neutron/plugins/vmware/nsx.ini`):

NSX-T:

```
nsx.ini
[nsx_v3]
metadata_proxy = bb29ca7c-f40a-44fa-bc7f-e9067eca98aa
dhcp_profile = 4a34be05-2984-426c-ac71-fcdad6397a83
native_metadata_route = 169.254.169.254/32
native_dhcp_metadata = True
nsx_api_password = Admin!23Admin
nsx_api_user = admin
nsx_api_managers = 10.162.16.55
default_tier0_router = 225cb6a6-6384-405c-9028-c531005ab187
default_overlay_tz = a613c2e5-a5c0-4827-90c3-559d26988645
default_edge_cluster_uuid =
4f5bcab7-ed8b-4f69-ae26-a55cc414bcdc
nova compute nova.conf:
[Neutron]
metadata_proxy_shared_secret = secret
service_metadata_proxy = True
ovs_bridge = nsx-managed
url = http://10.162.0.45:9696
region_name = RegionOne
auth_strategy = keystone
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = Neutron
auth_url = http://10.162.0.45/identity_v2_admin/v3
auth_type = password
```

Configuring NSX-T with Red Hat OpenStack Platform

On OverCloud Controller, modify Neutron to use NSX-T.

1. Modify neutron.conf.

```
[heat-admin@overcloud-controller-0 ~]$ sudo vi
/etc/neutron/neutron.conf
[DEFAULT]
core_plugin = vmware_nsx.plugin.NsxV3Plugin
#service_plugins=router,qos,trunk

allow_overlapping_ips = True

notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True

# If DHCP and Metadata Proxy offered by NSX-T (from
OpenStack Newton)
dhcp_agent_notification = False

# If L2 bridging (L2 Overlay/VLAN bridging)
#service_plugins = vmware_nsx_l2gw

# If LBaaS service
#service_plugins =
neutron_lbaas.services.loadbalancer.plugin.LoadBalancerPlugi
nv2
```

2. Create a plugins/vmware folder.

```
[heat-admin@overcloud-controller-0 ~]$ sudo mkdir
/etc/neutron/plugins/vmware
```

3. Create the nsx.ini file.

```
[heat-admin@overcloud-controller-0 ~]$ sudo
vi/etc/neutron/plugins/vmware/nsx.ini
```

4. Update the plugin.ini file to point to plugins/nsx.ini instead of plugins/ml2.ini.

```
[heat-admin@overcloud-controller-0 ~]$ sudo rm
/etc/neutron/plugin.ini
```

```
[heat-admin@overcloud-controller-0 ~]$ sudo ln -s
/etc/neutron/plugins/vmware/nsx.ini /etc/neutron/plugin.ini
```

On Overcloud Controller, install NSX-T Stable/Newton.

1. Copy the NSX-T plugin.

```
[heat-admin@overcloud-controller-0 ~]$ cd
```

```
[heat-admin@overcloud-controller-0 ~]$ git clone
https://github.com/openstack/vmware-nsx
```

2. Select the version for OpenStack Newton.

```
[heat-admin@overcloud-controller-0 ~]$ cd vmware-nsx
```

```
[heat-admin@overcloud-controller-0 ~]$ git checkout
stable/newton
```

3. Install the NSX-T plugin.

```
[heat-admin@overcloud-controller-0 ~]$ sudo python setup.py
install
```

4. Upgrade db schema.

```
[heat-admin@overcloud-controller-0 ~]$ sudo
neutron-db-manage upgrade heads
```

5. If SELinux is enabled, enable Neutron service to establish connections to NSX.

```
sudo setsebool -P neutron_can_network 1
```

On Overcloud Controller, restart Neutron and validate it is running.

1. Restart Neutron server.

```
sudo service neutron-server restart
```

2. Validate Neutron is active. If there are issues, view the log file at: /var/log/neutron/server.log

```
service neutron-server status

Redirecting to /bin/systemctl status  neutron-server.service

neutron-server.service - OpenStack Neutron Server

Loaded: loaded
(/usr/lib/systemd/system/neutron-server.service; enabled;
vendor preset: disabled)
```

```
Active: active (running) since Wed 2017-04-12 16:40:00 UTC;
1min 48s ag
```

On Overcloud Compute, install NSX-T.

1. Register OverCloud Compute VM on RedHat.

```
[heat-admin@overcloud-compute-0 ~]$ sudo
subscription-manager register --username <username>
--password <password> --auto-attach
```

2. Install the NSX-T requirements for RHEL.

```
sudo yum groupinstall "Virtualization Hypervisor"

sudo yum groupinstall "Virtualization Client"

sudo yum groupinstall "Virtualization Platform"

sudo yum groupinstall "Virtualization Tools"

sudo yum install c-ares

curl -O
http://apt.nicira.eng.vmware.com/apt/dropkick/rhel72/x86_64/
protobuf-python-2.5.0-8.el7.x86_64.rpm

curl -O
http://apt.nicira.eng.vmware.com/apt/dropkick/rhel72/x86_64/
leveldb-1.12.0-5.el7.x86_64.rpm

curl -O
http://apt.nicira.eng.vmware.com/apt/dropkick/rhel73/x86_64/
libev-4.15-6.el7.x86_64.rpm

curl -O
http://apt.nicira.eng.vmware.com/apt/dropkick/rhel73/x86_64/
python-gevent-1.0-2.el7.x86_64.rpm

sudo yum install *.rpm
```

3. Install NSX-T for RHEL.

```
tar -xvf nsx-lcp-2.0.0.0.0.5296168-rhel73_x86_64.tar.gz

cd nsx-lcp-rhel73_x86_64/

sudo yum install *.rpm


sudo /etc/init.d/openvswitch force-reload-kmod

sudo vi /etc/nova/nova.conf
sudo nsxcli
```