
networking-dpm Documentation

Release 0.0.1.dev56

OpenStack Foundation

Feb 14, 2017

Contents

1	Overview	3
2	Using networking-dpm	5
2.1	Installation	5
2.2	Configuration	6
3	Contributing to the project	11
3.1	Contributing	11
3.2	Specs	11
4	Links	13

This project provides the OpenStack Neutron mechanism driver and L2 agent for the PR/SM hypervisor of IBM z Systems and IBM LinuxOne machines that are in the DPM (Dynamic Partition Manager) administrative mode.

The DPM mode enables dynamic capabilities of the firmware-based PR/SM hypervisor that are usually known from software-based hypervisors, such as creation, deletion and modification of partitions (i.e. virtual machines) and virtual devices within these partitions, and dynamic assignment of these virtual devices to physical I/O adapters.

The Neutron mechanism driver and L2 agent for DPM components are needed on OpenStack compute nodes for DPM, along with the Nova virtualization driver for DPM.

For details about OpenStack for DPM, see the [documentation of the nova-dpm OpenStack project](#).

CHAPTER 1

Overview

2.1 Installation

The networking-dpm package provides two components:

- Neutron mechanism driver for DPM
- L2 agent for DPM

The Neutron mechanism driver for DPM must be registered with the Neutron server on the OpenStack controller node.

The L2 agent for DPM must be installed on every OpenStack compute node for DPM.

This section describes the manual installation of these components onto a controller node and compute node that have already been installed by some means.

The networking-dpm package is released on PyPI as package [networking-dpm](#).

The following table indicates which version of the networking-dpm package on PyPI to use for a particular OpenStack release:

OpenStack release	networking-dpm version
Ocata	1.x.x

Typically, the networking-dpm package will increase its major version number by one for each new OpenStack release.

If you want to install the package for a particular OpenStack release, it is recommended to use the package that has been released to PyPI, rather than installing from a particular branch of a Git repository.

To do that, identify the major version number for the desired OpenStack release from the table above, and install the latest minor and fix version of the package for that major version, also specifying the global upper constraints file for the desired OpenStack release (the latter ensures that you get the right versions of any dependent packages).

For example, for Ocata:

```
$ constraints_file=https://git.openstack.org/cgit/openstack/requirements/plain/upper-  
→constraints.txt?h=stable/ocata  
$ pip install -c$constraints_file "networking-dpm >=1,<2"
```

If you have good reasons to install the latest not yet released fix level of the networking-dpm package for a particular (released) OpenStack release, install the networking-dpm package from the stable branch of the GitHub repo for that OpenStack release:

For example, for Ocata:

```
$ constraints_file=https://git.openstack.org/cgit/openstack/requirements/plain/upper-  
→constraints.txt?h=stable/ocata  
$ pip install -c$constraints_file git+https://git.openstack.org/openstack/networking-  
→dpm@stable/ocata
```

If you are a developer and want to install the latest code of the networking-dpm package for the OpenStack release that is in development:

```
$ constraints_file=https://git.openstack.org/cgit/openstack/requirements/plain/upper-  
→constraints.txt?h=master  
$ pip install -c$constraints_file git+https://git.openstack.org/openstack/networking-  
→dpm@master
```

The pip commands above install the packages into the currently active Python environment.

If your active Python environment is a virtual Python environment, the commands above can be issued from a userid without sudo rights.

If you need to install the packages into the system Python environment, you need sudo rights:

```
$ sudo pip install ...
```

After installing the networking-dpm package, proceed with its *Configuration*.

Note that you will also need to install and configure the nova-dpm package on the compute node. For its documentation, see <http://nova-dpm.readthedocs.io/en/latest/>.

2.2 Configuration

2.2.1 Neutron DPM mechanism driver

The Neutron DPM mechanism driver itself does not require any DPM-specific configuration options.

But certain well known Neutron ML2 (Modular Layer 2) configuration options in the ML2 configuration file (typically *ml2_conf.ini*) are required in order to use it.

Enable the DPM mechanism driver

The *DPM* mechanism driver must be enabled using Neutron ML2 *mechanism_drivers* config option. Typically the DPM mechanism driver must be configured amongst other mechanisms (like *OVS*) that are used on the network node or compute nodes managing other hypervisor types (like *KVM*). The following example enables the *ovs* and the *DPM* mechanism driver in Neutrons ML2 config file *ml2_conf.ini*:

```
[DEFAULT]  
mechanism_drivers = ovs,dpm
```

More details can be found in the OpenStack Configuration Reference on docs.openstack.org.

Configure ML2 for self service VLAN networks

For making use of self service VLAN networks, the physical networks and their corresponding VLAN ranges must be defined in the Neutron servers ML2 configuration file:

```
[ml2_type_vlan]
network_vlan_ranges = <physical_network>:<vlan_min>:<vlan_max>,<physical_network2>:
↪<vlan_min>:<vlan_max>
```

More details can be found in the OpenStack Configuration Reference on docs.openstack.org.

Note: Each physical network that should be used for self service VLAN networks must also be defined in the Neutron DPM agent *physical_adapter_mappings* configuration option. This is also true for other Neutron L2 Agents (for example the Neutron Open vSwitch Agent). They all have a similar *mappings* option that must be configured accordingly.

Note: This config option is not required if only VLAN provider networks will be used.

2.2.2 Neutron DPM agent

The Neutron DPM agent on the compute node requires DPM-specific options. But also some well known Neutron options can be set.

General Neutron options

The following common Neutron options can be set in the Neutron DPM agent's configuration file:

- [default] host
- [agent] quitting_rpc_timeout
- [agent] polling_interval

More details can be found in the OpenStack Configuration Reference on docs.openstack.org.

DPM-specific options

Those are the DPM-specific configuration options required by the Neutron DPM agent.

Note: This configuration is auto-generated from the networking-dpm project when this documentation is built. So if you are having issues with an option, please compare your version of networking-dpm with the version of this documentation.

The sample configuration can also be viewed in file form.

```
[DEFAULT]

[dpm]
#
# Configuration options for IBM z Systems and IBM LinuxONE in DPM (Dynamic
```

```
# Partition Manager) administrative mode. A z Systems or LinuxONE machine is
# termed "CPC" (Central Processor Complex). The CPCs are managed via the Web
# Services API exposed by the "HMC" (Hardware Management Console). One HMC can
# manage multiple CPCs.
#
#
# DPM config options for the Neutron agent on the compute node (one agent
# instance for each OpenStack hypervisor host) specify the target CPC, the HMC
# managing it, and the OpenStack physical networks for the OpenStack hypervisor
# host and their backing network adapters and ports in the target CPC.
#
# From networking_dpm
#
#
# The OpenStack physical networks that can be used by this OpenStack hypervisor
# host, and their backing network adapters and ports in the target CPC.
#
# This is a multi-line option. Each instance (line) of the option defines one
# physical network for use by this OpenStack hypervisor host, and the network
# adapter and port that is used for that physical network, using this syntax:
#
#   <physical-network-name>:<adapter-object-id>[:<port-element-id>]
#
# * <physical-network-name>` is the name of the OpenStack physical network.
# * <adapter-object-id>` is the object-id of the network adapter in the target
#   CPC that is used for this physical network.
# * <port-element-id>` is the element-id of the port on that network adapter.
#   It is optional and defaults to 0.
#
# The instances (lines) of this option for a particular Neutron agent
#
# * must not specify the same physical network more than once, and
# * must not specify the same adapter and port more than once, and
# * must have all of their physical networks specified in the
#   corresponding `*mappings` config option of the Neutron L2 agent service
#   on all network nodes, and
# * must have all of their physical networks specified in the
#   `ml2.network_vlan_ranges` config option of the Neutron server, if vlan
#   self service networks should be used.
# (multi valued)
#physical_network_adapter_mappings = physnet1:12345678-1234-1234-1234-123456789a
#physical_network_adapter_mappings = physnet2:12345678-1234-1234-1234-123456789b:1
#physical_network_adapter_mappings = physnet3:12345678-1234-1234-1234-123456789c:0
#
#   Hostname or IP address of the HMC that manages the target CPC (string
#   value)
#hmc = <None>
#
#   User name for connection to the HMC (string value)
#hmc_username = <None>
#
```

```
#      Password for connection to the HMC (string value)
#hmc_password = <None>

#
#      DPM Object-id of the target CPC (string value)
#cpc_object_id = <None>
```

Contributing to the project

3.1 Contributing

If you would like to contribute to the development of the networking-dpm project, you must follow the rules for OpenStack contributions described in the “If you’re a developer, start here” section of this page:

<http://wiki.openstack.org/HowToContribute>

Once those steps have been completed, changes to the networking-dpm project should be submitted for review via the Gerrit tool, following the workflow documented at:

<http://wiki.openstack.org/GerritWorkflow>

Pull requests submitted through GitHub will be ignored.

The Git repository for the networking-dpm project is here:

<http://git.openstack.org/cgit/openstack/networking-dpm>

Bugs against the networking-dpm project should be filed on Launchpad (not on GitHub):

<https://bugs.launchpad.net/networking-dpm>

Pending changes for the networking-dpm project can be seen on its Gerrit page:

<https://review.openstack.org/#/q/project:openstack/networking-dpm>

3.2 Specs

3.2.1 Ocata

(Placeholder Spec)

This file is just a placeholder for Ocata specs directory. It will be removed soon after some spec for Ocata is merged.

The latest spec template is found at `specs/template.rst` in the `networking-dpm` repository.

Problem Description

Sphinx toctree complains if no file exists in a directory specified in toctree glob.

Proposed Change

Add this file.

References

None.

CHAPTER 4

Links

- Documentation: <http://networking-dpm.readthedocs.io/en/latest/>
- Source: <http://git.openstack.org/cgit/openstack/networking-dpm>
- Github shadow: <https://github.com/openstack/networking-dpm>
- Bugs: <http://bugs.launchpad.net/networking-dpm>
- Gerrit: <https://review.openstack.org/#/q/project:openstack/networking-dpm>