
Fuel VMware DVS Plugin Guide

Release 3.1-3.1.0-1

Mirantis Inc.

October 05, 2016

1	Overview	1
1.1	Introduction	1
1.2	Key terms	1
1.3	Requirements	2
1.4	Prerequisites	2
1.5	Limitations	3
1.6	Release notes	3
1.7	Licenses	3
1.8	Useful links	3
2	Install and configure VMware DVS plugin for Fuel	4
2.1	Install VMware DVS plugin	4
2.2	Uninstall VMware DVS plugin	5
2.3	Configure an environment with VMware DVS plugin	6
2.4	Verify a deployed environment with VMware DVS plugin	10
3	Use VMware DVS plugin for Fuel	11
3.1	Using advanced Neutron possibilities with VMware DVS plugin	11
3.2	Troubleshooting	13

OVERVIEW

1.1 Introduction

The purpose of this document is to describe how to install, configure, and use the VMware DVS plugin 3.1.0 for Fuel 9.1.

Mirantis OpenStack supports using vCenter as a hypervisor on vCenter-only or heterogeneous environments that are mixed with KVM. The `vmware_dvs` driver for Neutron ML2 plugin allows using Neutron for networking in such environments. Therefore, you get the following advanced network features for your environment:

- Create multi-tier networks (for example, web tier, database tier, application tier)
- Control over IP addressing and security groups' rules
- Add and configure custom services (for example, firewall, intrusion-prevention system)
- Use VPN/bridge to a remote physical hosting or customer premises

1.2 Key terms

The table below lists the key terms, acronyms, and abbreviations that are used in this document.

Term/abbreviation	Definition
Neutron ML2 plugin	The Neutron Modular Layer 2 plugin. It is a framework allowing OpenStack Networking to simultaneously utilize the variety of Layer 2 networking technologies.
OVS	Open vSwitch
VDS	VMware vSphere® Distributed Switch
VM	Virtual machine
vmware_dvs driver	The driver in the Neutron ML2 plugin which provides interaction with dvSwitch on vCenter.
VMware DVS plugin	The plugin for Fuel that installs and configures <code>vmware_dvs</code> driver on a Mirantis OpenStack environment.
VMware ESXi	Bare-metal hypervisor
VMware vCenter server	Central control point for VMware vSphere
VMware vSphere	VMware cloud computing virtualization operating system

1.3 Requirements

The VMware DVS plugin 3.1.0 for Fuel has the following requirements:

- Fuel 9.1 on Mitaka
- vCenter 5.5/6.0

1.4 Prerequisites

Before you install and start using the VMware DVS plugin on Fuel, complete the following steps:

1. Install and set up [Fuel 9.1](#).
2. Plan the vSphere integration. For details, see [Mirantis OpenStack Planning Guide](#).

See also:

- [VMware vSphere 5.5 official documentation](#)
- [VMware vSphere in OpenStack Configuration Reference](#)

3. Create a [vCenter service account](#).
4. In the vCenter service account, apply the following minimum privileges for *Distributed switch* and *dvPort group*:

Permission	Privilege
dvSwitch	<ul style="list-style-type: none"> • Port configuration operation
dvPort Group	<ul style="list-style-type: none"> • dvPort group.Create • dvPort group.Delete • dvPort group.Modify • dvPort group.Policy operation

This allows the VMware DVS plugin to use manipulation resources of VMware vSphere Distributed Switch (VDS).

5. Create and properly configure VDSes on vCenter that will be used for your environment. For details, see the VDS videos in the *Technical Details -> Resources* section on the [VMware Distributed Switch](#) page.
6. Connect the VMware DVS plugin to the pre-created and configured VDSes.

Note: The VMware DVS plugin does not create new VDSes but uses the existing ones.

1.5 Limitations

The VMware DVS plugin 3.1.0 for Fuel has the following limitations:

- The plugin is enabled only on environments with Neutron as the networking option.
- Only VLANs are supported for the tenant network separation.
- Only vSphere versions 5.5 and 6.0 are supported.
- IPv6 is not supported.
- Neutron Distributed Virtual Routers (DVR) feature is not supported.
- Each vSphere cluster should be connected to one (and only one) VDS.

1.6 Release notes

The VMware DVS plugin 3.1.0 contains the following updates:

- Added support of multiple uplinks on VDS.

1.7 Licenses

Component	License
vmware_dvs driver	Apache 2.0
VMware DVS plugin	Apache 2.0

1.8 Useful links

For more information about Fuel VMware DVS plugin described in this document, see:

- [Specification](#)
- [GitHub project](#)
- [Launchpad project](#)

INSTALL AND CONFIGURE VMWARE DVS PLUGIN FOR FUEL

2.1 Install VMware DVS plugin

Before you proceed with the VMware DVS plugin installation, verify that:

1. You have completed the *Prerequisites* steps.
2. All the nodes of your future environment are *DISCOVERED* on the Fuel Master node.
3. You have connectivity to correctly configured vCenter with VDSes and clusters created.

To install the VMware DVS plugin:

1. Download the VMware DVS plugin from the [Fuel Plugin Catalog](#).
2. Copy the plugin .rpm package to the Fuel Master node:

```
$ scp fuel-plugin-vmware-dvs-3.1-3.1.0-1.noarch.rpm <Fuel Master node ip>:/tmp
```

3. Log in to the Fuel Master node CLI as root.
4. Install the plugin:

```
# fuel plugins --install /tmp/fuel-plugin-vmware-dvs-3.1-3.1.0-1.noarch.rpm
```

5. Verify that the plugin was installed successfully:

```
# fuel plugins
```

```
+-----+-----+-----+-----+
| id    | name                               | version | package_version |
+-----+-----+-----+-----+
| 2     | fuel-plugin-vmware-dvs             | 3.1.0   | 4.0.0           |
+-----+-----+-----+-----+
```

6. Proceed to *Configure an environment with VMware DVS plugin*.

2.2 Uninstall VMware DVS plugin

To uninstall VMware DVS plugin, follow the steps below:

1. Log in to the Fuel Master node CLI.
2. Delete all the environments in which VMware DVS plugin is enabled:

```
# fuel --env <ENV_ID> env delete
```

3. Uninstall the plugin:

```
# fuel plugins --remove fuel-plugin-vmware-dvs==3.1.0
```

4. Verify whether the VMware DVS plugin was uninstalled successfully:

```
# fuel plugins
```

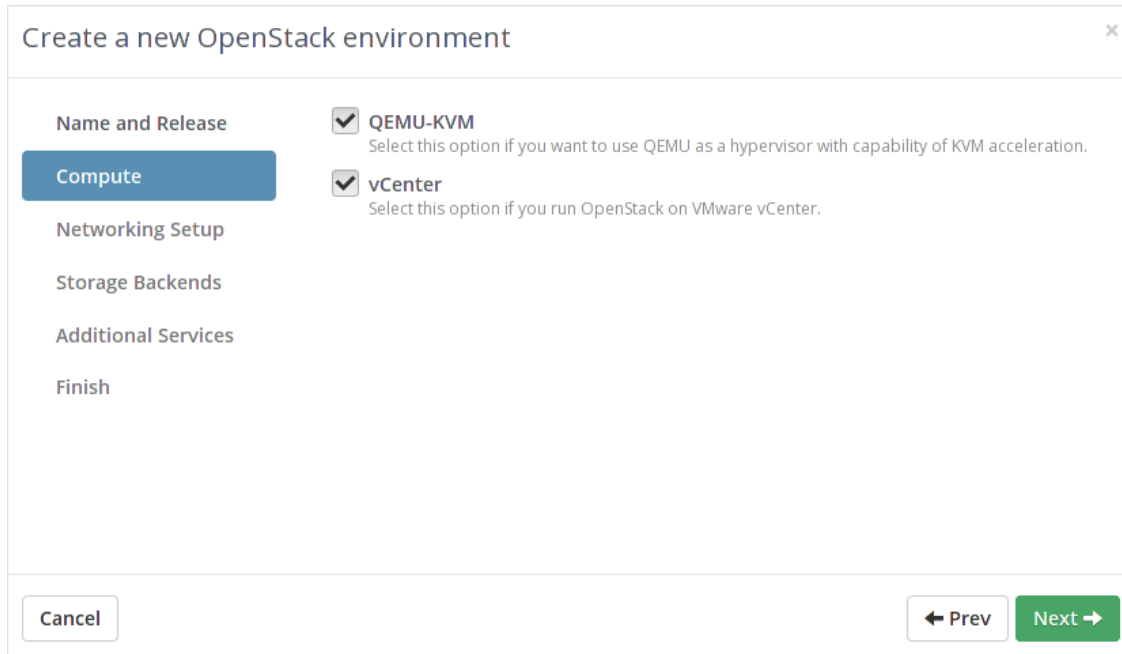
The VMware DVS plugin should not appear in the output list.

2.3 Configure an environment with VMware DVS plugin

Configuring and deploying an environment with VMware DVS plugin involves creating an environment in Fuel and modifying the environment settings.

To configure an OpenStack environment with VMware DVS plugin:

1. Using the Fuel web UI, follow steps 1 to 5 of the [Create a new OpenStack environment](#) instruction.
2. In the *Compute* menu, select *vCenter*:



The screenshot shows a web-based configuration wizard titled "Create a new OpenStack environment". On the left side, there is a vertical menu with the following steps: "Name and Release", "Compute" (which is highlighted with a blue background), "Networking Setup", "Storage Backends", "Additional Services", and "Finish". On the right side, under the "Compute" step, there are two checked options:

- QEMU-KVM**
Select this option if you want to use QEMU as a hypervisor with capability of KVM acceleration.
- vCenter**
Select this option if you run OpenStack on VMware vCenter.

At the bottom of the wizard, there are three buttons: "Cancel" on the left, "← Prev" in the middle, and "Next →" on the right.

3. In the *Networking Setup* menu, select *Neutron with VMware DVS*:

Create a new OpenStack environment

Name and Release

Compute

Networking Setup

Storage Backends

Additional Services

Finish

- Neutron with ML2 plugin** ✓
Framework that enables simultaneous utilization of the layer 2 networking technologies through drivers.
- Neutron with VLAN segmentation** ⓘ
Your network hardware must be configured for VLAN segmentation. This option supports up to 4095 networks.
- Neutron with tunneling segmentation** ⚠
By default VXLAN tunnels will be used. This option supports millions of tenant data networks.
- Neutron with VMware DVS** ✓
Neutron with VMware DVS ML2 plugin

Cancel ← Prev Next →

4. Follow steps 8-10 of the *Create a new OpenStack environment* instruction.

5. In the *Nodes* tab of the Fuel web UI, add at least one *Controller* node to the environment:

Cancel Apply Changes

Sort By Status ↓

Assign Roles

- Controller**
The Controller initiates orchestration activities and provides an external API. Other components like Glance (image storage), Keystone (identity management), Horizon (OpenStack dashboard) and Nova-Scheduler are installed on the controller as well.

(Optional) You can also add one dedicated *Compute VMware* node:

- Compute VMware**
A node that runs nova-compute with VCDriver, that manages ESXi computing resources via VMware vCenter.

6. In the *Networks* tab, click *Other*:

- (a) Select the *Neutron VMware DVS ML2 plugin* checkbox.
- (b) Specify the *Cluster to VDSwitch mapping*. Please notice that in the 3.1 release it has new format:
 - i. New string is used as a delimiter between clusters.
 - ii. There are 2 new columns: list of active uplinks and list of standby uplinks. Both are optional.
 - iii. The semicolon is used as a delimiter between uplinks.
 - iv. There is no limitation for amount of uplinks.
 - v. Thereby there are next options for a mapping-string:

A. ClusterName:VDSName:ActiveUplink1;ActiveUplink2:StandbyUplink1;StandbyUplink2

B. ClusterName:VDSName:ActiveUplink1;ActiveUplink2;...;ActiveUplinkN

C. ClusterName:VDSName

vi. There is no option to set standby uplinks without active uplinks.

vii. All uplinks should be presented on real VDS.

(c) If you want to use security groups on your ports, select *Use the VMware DVS firewall driver*.

Neutron VMware DVS ML2 plugin

Versions 3.1.0

Use the VMware DVS firewall driver

Enter the Cluster to dvSwitch mapping.

Cluster1:Cluster1:dvUplink1:dvUplink2:dvUplink3
Cluster2:Cluster2:dvUplink1:dvUplink2:dvUplink3

List of strings with format
Cluster:VDS:Active1;Active2:Standby1;Standby2 where Active and Standby is active and standby uplinks on a VDS.

See the [Teaming and Failover Policy](#) for more detail about uplinks usage on VDS.

Caution: The VMware DVS ML2 plugin does not support the Distributed Virtual Routers (DVR) feature. Therefore, do not select *Neutron DVR* in *Neutron L3 Configuration* -> *Neutron Advanced Configuration*.

7. In the *VMware* tab, fill in the VMware configuration fields:

The screenshot shows the VMware vCenter Settings page. At the top, there is a navigation bar with tabs for Dashboard, Nodes, Networks, Settings, VMware, Logs, History, Workflows, and Health Check. The main heading is "VMware vCenter Settings" followed by "vCenter". Below this, there are several configuration fields:

- Availability zone:** A text input field containing "vcenter". To its right is the label "Availability zone name".
- vCenter host:** A text input field containing "172.16.0.145". To its right is the label "vCenter host or IP".
- vCenter username:** A text input field containing "root". To its right is the label "vCenter username".
- vCenter password:** A password input field with masked characters (dots) and a toggle icon. To its right is the label "vCenter password".
- Bypass vCenter certificate verification:** A checked checkbox.
- CA file:** A file selection field showing "No file selected" and a file icon. To its right is a descriptive text: "File containing the trusted CA bundle that emitted vCenter server certificate. Even if CA bundle is not uploaded, certificate verification is turned on."

Nova Computes

+ - Nova Compute Instance

The screenshot shows the Nova Compute Instance configuration page. It contains the following fields:

- vSphere cluster:** A text input field containing "Cluster1". To its right is the label "vSphere cluster".
- Service name:** A text input field containing "sn1". To its right is the label "Service name".
- Datastore regex:** A text input field containing ".*". To its right is the label "Datastore regex".
- Target node:** A dropdown menu with "controllers" selected. To its right is the label "Target node for nova-compute service".

8. Make additional configuration adjustments.
9. Proceed to the environment deployment.

2.4 Verify a deployed environment with VMware DVS plugin

After you deploy an environment with VMware DVS plugin, complete the following verification steps:

1. Log in to a controller node.
2. Run the **neutron agent-list** command to verify whether the DVS agent is present in the list of Neutron agents and is ready for use:
 - The `alive` column should contain the `:~)` value.
 - The `admin_state_up` column should contain the `True` value.

```
$ neutron agent-list
+-----+-----+-----+-----+-----+-----+
|id   |agent_type|host           |alive |admin_state_up|binary           |
+-----+-----+-----+-----+-----+-----+
|...  |DVS agent |vcenter-sn2|:~)   |True         |neutron-dvs-agent|
+-----+-----+-----+-----+-----+-----+
```

Note: In the example above, the `availability_zone` column was removed from the output of the **neutron agent-list** command.

3. Log in to the Fuel web UI.
4. Click the *Health Check* tab.
5. Run necessary health tests. For details, see: [Verify your OpenStack environment](#).

USE VMWARE DVS PLUGIN FOR FUEL

3.1 Using advanced Neutron possibilities with VMware DVS plugin

Once you deploy an OpenStack environment with VMware DVS plugin, you can start using Neutron for networking. When Neutron creates a new network, it does not affect any VDS until a port in that network is attached to a VM launched on a corresponding environment.

DVS security groups

The functionality of the VMware DVS plugin security groups differs from the KVM implementation:

- The VMware DVS plugin does not support stateful firewall properties and ICMP types. It realizes the emulation logic to support similar behavior. The plugin installs a reverse traffic rule for each security group rule.
- The VMware DVS plugin state emulation logic uses ephemeral port range filter to rise the security of reverse rules implementation.

You can only add the reverse rules to correctly launch an EC2-compatible image with metadata request and DNS access.

To add reverse rules:

1. Implement a custom TCP egress rule to 169.254.169.254//32 CIDR port 80.
2. Implement a custom UDP egress rule to <DNS server IP or 0.0.0.0//0> CIDR port 53.

The VMware DVS plugin installs four rules:

1. TCP egress from any IP ports 32768-65535 to metadata IP port 80
2. TCP ingress from metadata IP port 80 to any IP ports 32768-65535
3. UDP egress from any IP ports 32768-65535 to DNS IP port 53
4. UDP ingress from DNS IP port 53 to any IP ports 32768-65535

Note: 32768-65535 is a useful ephemeral port range for most Linux kernels and Windows hosts.

Example of a common egress TCP rule:

- TCP egress to any ports 0.0.0.0/0 CIDR

It works as follows:

- TCP egress from any IP ports 32768-65535 to any IP any port
- TCP ingress from any IP any port to any IP ports 32768-65535

Once applied, the private ports of your VM, such as HTTP or SSH, will be closed.

The VMWare DVS plugin supports only symmetric ICMP interaction. If your host can ping a destination host, it means that the destination host can ping your host by reverse rules.

3.2 Troubleshooting

This section contains a guidance on how to ensure that the VMware DVS plugin is up and running on your deployed environment.

To find logs

The VMware DVS driver consists of two parts: the mechanism driver of Neutron and the VMware DVS agent. Therefore, two main sources of information for troubleshooting are:

- /var/log/neutron/server.log
- /var/log/neutron/vmware-dvs-agent-....log

To verify Neutron configuration files

To deliver a stable performance of the VMware DVS plugin, verify that the Neutron configuration files contain the following values:

- /etc/neutron/neutron.conf:


```
notification_driver=messagingv2
```
- /etc/neutron/plugin.ini:


```
[m12]
mechanism_drivers =openvswitch,l2population,vmware_dvs
[m12_vmware]
vsphere_login=<vsphere_user>
vsphere_hostname=<vsphere_ip>
vsphere_password=<vsphere_password>
```
- /etc/neutron/plugins/ml2/vmware_dvs-<vcenter AZ>-<service name>.ini:


```
[DEFAULT]
host=<vcenter AZ>-<service name>

[securitygroup]
enable_security_group = True
firewall_driver=mech_vmware_dvs.agentDVS.vCenter_firewall.DVSFirewallDriver

[m12_vmware]
vsphere_login=<vsphere_user>
network_maps=physnet2:<VDS>
vsphere_hostname=<vsphere_ip>
vsphere_password=<vsphere_password>
```

To verify neutron-dvs-agent services

All neutron-dvs-agent services should run on the corresponding nodes:

- On controllers: p_neutron_plugin_vmware_dvs_agent_\${host} in Corosync
- On compute-vmware: neutron-plugin-vmware-dvs-agent-\${host} in the init script

To verify connectivity

Check the connectivity between controller nodes and vCenter using the **ping** command.