
Fuel VMware DVS plugin user guide

Release 2.1-2.1.0-1

Mirantis Inc.

June 27, 2016

1	Introduction	1
1.1	Key terms, acronyms and abbreviation	1
1.2	The VMware DVS plugin	1
1.3	Licensing information	1
1.4	Prerequisites	2
1.5	Requirements	2
1.6	Limitations	2
1.7	Release notes	2
2	Installation and create an environment	3
2.1	Installing the VMware DVS plugin	3
2.2	Removing the VMware DVS plugin	4
2.3	Create and Configure an environment with VMware DVS plugin	4
3	User Guide	8
3.1	Using advanced neutron's possibilities with VMware DVS plugin	8
3.2	Verification	10
3.3	Troubleshooting	10

INTRODUCTION

1.1 Key terms, acronyms and abbreviation

Term/abbreviation	Definition
VM	Virtual Machine
MOS	Mirantis OpenStack
OVS	Open vSwitch
Neutron ML2 plugin	The Neutron Modular Layer 2 plugin is a framework allowing OpenStack Networking to simultaneously utilize the variety of layer 2 networking technologies
vmware_dvs driver	The driver in the Neutron ML2 plugin which provides interaction with dvSwitch on vCenter
VMware DVS plugin	The plugin for Fuel which installs and configures vmware_dvs driver on a MOS environment
dvSwitch	VMware vSphere® Distributed Switch
VMware ESXi	bare-metal hypervisor
VMware vCenter Server	Central control point for VMware vSphere
VMware vSphere	VMware's cloud computing virtualization operating system

1.2 The VMware DVS plugin

MOS supports using vCenter as a hypervisor in a vCenter-only or heterogeneous, mixed with KVM environments. There is the vmware_dvs driver for Neutron ML2 plugin which provides usage Neutron for networking in such environments. Thereby environments receives an advanced network features:

- Ability to create multi-tier networks (e.g., web tier, db tier, app tier).
- Control over IP addressing and security groups's rules.
- Ability to insert an configure their own services (e.g., firewall, IPS)
- VPN/Bridge to remote physical hosting or customer premises.

1.3 Licensing information

Component	License
vmware_dvs driver	Apache 2.0
VMware DVS plugin	Apache 2.0

1.4 Prerequisites

The VMware DVS plugin works via manipulation resources of a VMware vSphere Distributed Switch. It means that it has to have connectivity to precreated and **well configured** dvSwitch on the vCenter which will be used in this environment. Only dvSwitch v5.5 or later supported

1.5 Requirements

The plugin has the following requirements for software:

Requirement	Version
Fuel	8.0
vCenter	5.5/6.0

1.6 Limitations

- Only VLANs are supported for tenant network separation.
- Only vSphere 5.5 & 6.0 are supported.
- All hosts of each vSphere Cluster should be connected with dvSwitch with same name as that Cluster.
- There is no IPv6 support.

1.7 Release notes

Release notes for VMware DVS plugin 2.1.0:

- New architecture with the vmware-dvs neutron agent.
- A name of dvSwitches is limited by 44 symbols.
- Support for remote security groups.
- Eventually consistent security group engine.
- Enhancement of stability especially in case of multiple simultaneously runned tasks.
- The performance improvement.

INSTALLATION AND CREATE AN ENVIRONMENT

2.1 Installing the VMware DVS plugin

Make sure that:

- you have the installed the [Fuel Master node](#)
- all the nodes of your future environment are discovered and functional.
- there is a connectivity to correctly configured vCenter with VDS'es and clusters created. Please, see the [Mirantis OpenStack Planning Guide](#), [User Guide](#) and [this plugin's specification](#) for information on configuring vCenter.

1. Download the plugin from the [Fuel Plugin Catalog](#).

2. Copy the plugin into Fuel Master node:

```
$ scp fuel-plugin-vmware-dvs-2.1-2.1.0-1.noarch.rpm <Fuel Master node ip>:/tmp
```

3. Log into the Fuel Master node and install the plugin:

```
$ ssh root@<Fuel Master node ip>
[root@nailgun ~]# fuel plugins --install /
/tmp/fuel-plugin-vmware-dvs-2.1-2.1.0-1.noarch.rpm
[root@nailgun ~]# fuel plugins
DEPRECATION WARNING: /etc/fuel/client/config.yaml exists and will
be used as the source for settings. This behavior is deprecated.
Please specify the path to your custom settings file in the
FUELCLIENT_CUSTOM_SETTINGS environment variable.
```

```
+-----+-----+-----+-----+
| id    | name                | version | package\_version |
+-----+-----+-----+-----+
| 2     | fuel-plugin-vmware-dvs | 2.1.0   | 4.0.0            |
+-----+-----+-----+-----+
```

2.2 Removing the VMware DVS plugin

To uninstall VMware DVS plugin, follow these steps:

1. Delete all environments in which VMware DVS plugin has been enabled.
2. Uninstall the plugin:

```
# fuel plugins --remove fuel-plugin-vmware-dvs==2.1.0
```

3. Check if the plugin was uninstalled successfully:

```
+-----+-----+-----+-----+
| id   | name   | version | package_version |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

2.3 Create and Configure an environment with VMware DVS plugin

1. Create a new OpenStack environment with Fuel UI wizard.

Create a new OpenStack environment
×

Name and Release

Compute

Networking Setup

Storage Backends

Additional Services

Finish

Name

OpenStack Release

By default, Fuel uploads the software packages for the Fuel Slave nodes from the external repositories. Please verify the Fuel Master node has the Internet connection. If the Fuel Master node does not have access to the Internet, you must create a local mirror with all required software packages and configure Fuel to use the mirror before you deploy an OpenStack environment.

This option will install the OpenStack Liberty packages using Ubuntu as a base operating system. With high availability features built in, you are getting a robust, enterprise-grade OpenStack deployment.

Cancel

← Prev

Next →

2. In *Compute* menu, select *vCenter* checkbox:

Create a new OpenStack environment ×

<p>Name and Release</p> <p>Compute</p> <p>Networking Setup</p> <p>Storage Backends</p> <p>Additional Services</p> <p>Finish</p>	<p><input checked="" type="checkbox"/> QEMU-KVM Select this option if you want to use QEMU as a hypervisor with capability of KVM acceleration.</p> <p><input checked="" type="checkbox"/> vCenter Select this option if you run OpenStack on VMware vCenter.</p>
--	---

Cancel
← Prev
Next →

3. Select *Neutron with VMware DVS* for *Networking Setup*

Create a new OpenStack environment ×

<p>Name and Release</p> <p>Compute</p> <p>Networking Setup</p> <p>Storage Backends</p> <p>Additional Services</p> <p>Finish</p>	<p><input checked="" type="radio"/> Neutron with ML2 plugin ✓ Framework that enables simultaneous utilization of the layer 2 networking technologies through drivers.</p> <p><input checked="" type="checkbox"/> Neutron with VLAN segmentation ⓘ Your network hardware must be configured for VLAN segmentation. This option supports up to 4095 networks.</p> <p><input type="checkbox"/> Neutron with tunneling segmentation ⚠ By default VXLAN tunnels will be used. This option supports millions of tenant data networks.</p> <p><input checked="" type="checkbox"/> Neutron with VMware DVS ✓ Neutron with VMware DVS ML2 plugin</p>
--	---

Cancel
← Prev
Next →

4. Finish environment creation following [documentation](#).
5. Open the *Nodes* tab and **add** at least 1 Controller node to the environment:

(Optional) You can also add 1 dedicated Compute VMware node:

6. Open the *Networks* tab of the Fuel Web UI and chose the *Other* subtab. Select the *Neutron VMware DVS ML2 plugin* checkbox

Neutron VMware DVS ML2 plugin

Versions 2.1.0

Use the VMware DVS firewall driver

and set the checkbox “Use the VMware DVS firewall driver” if you want to use security groups on your ports. VMware DVS ML2 plugin does not support DVR feature. Keep Neutron DVR checkbox on Neutron Advanced Configuration tab at unchecked state.

7. Fill in the VMware configuration fields on the *VMware* tab:



VMware vCenter Settings

vCenter

Availability zone	<input type="text" value="vcenter"/>	Availability zone name
vCenter host	<input type="text" value="172.16.0.254"/>	vCenter host or IP
vCenter username	<input type="text" value="administrator@vsphere.local"/>	vCenter username
vCenter password	<input type="password" value="••••••••"/> <input type="button" value="👁"/>	vCenter password

Nova Computes

⊕ Nova Compute Instance

vSphere cluster	<input type="text" value="Cluster1"/>	vSphere cluster
Service name	<input type="text" value="sn1"/>	Service name
Datastore regex	<input type="text" value=".*"/>	Datastore regex
Target node	<input type="text" value="controllers"/>	Target node for nova-compute service

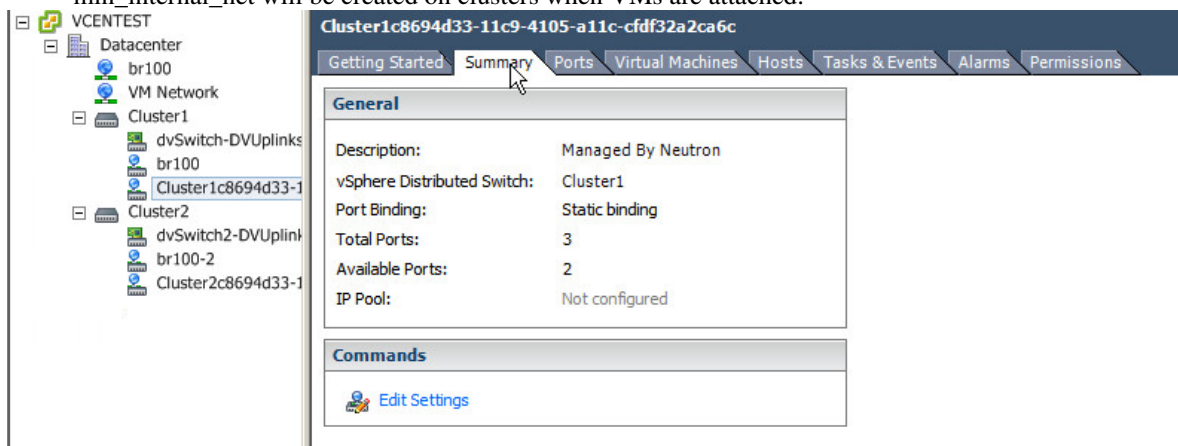
(Optional) Choose Compute VMware node if your environment has the role:

Target node	<input type="text" value="controllers"/> <ul style="list-style-type: none"> <li style="background-color: #f0f0f0; padding: 2px;">controllers <li style="background-color: #e67e22; color: white; padding: 2px;">controllers <li style="padding: 2px;">Untitled (c1:53) (28:c1:53) 	Target node for nova-compute service
-------------	--	--------------------------------------

- The rest of configuration is up to you. See [Mirantis OpenStack User Guide](#) for instructions.
- Click *Deploy changes* button to finish.

3.1 Using advanced neutron's possibilities with VMware DVS plugin

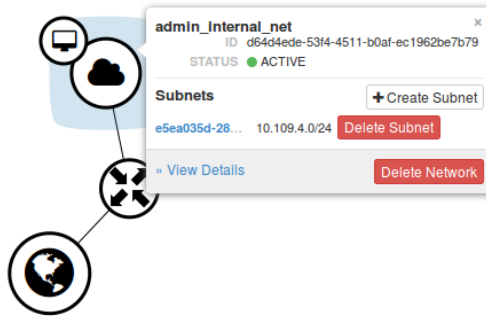
1. Once OpenStack has been deployed, we can start using Neutron for networking. The port group for admin_internal_net will be created on clusters when VMs are attached:



2. In Horizon, the network topology should look like:

Network Topology

Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.



where VMware is the name of the instance located on the vCenter.

3. You can use Neutron for such instance brand the same way as for KVM-located instances.
4. DvSwitch Security groups functionality differs from KVM implementation. VMWare dvSwitch does not support stateful firewall properties and ICMP types. DVS Plugin realises emulation logic to support the similar behavior. It installs reverse traffic rule for each SG rule. VMWare DVS plugin state emulation logic uses ephemeral port range filter to rise security of reverse rules implementation.

Usage of Remote Security Groups is possible but it can cause rules changes for multiple ports. SG engine will change rules in eventually consistent manner. Do not use Remote Security Groups with many attached ports if you need fast security rules changes.

Just add only those rules if you want to correctly launch EC2 compatible image with metadata request and DNS access:

Implement Custom TCP Egress rule to 169.254.169.254/32 CIDR port 80
 Implement Custom UDP Egress rule to '<DNS server IP or 0.0.0.0/0>' CIDR port 53

DVS plugin will install four rules:

- (a) TCP Egress from any IP ports 32768-65535 to metadata IP port 80
- (b) TCP Ingress from metadata IP port 80 to any IP ports 32768-65535
- (c) UDP Egress from any IP ports 32768-65535 to DNS IP port 53
- (d) UDP Ingress from DNS IP port 53 to any IP ports 32768-65535

32768-65535 is the useful ethemetal port range for most Linux kernels and Windows hosts.

Common egress TCP rule looks like this:

TCP Egress to any ports 0.0.0.0/0 CIDR

It works like:

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

and private ports of your VM like http or ssh will be closed.

DVS plugin support only symmetric ICMP interaction. If your host can ping destination host, it means the destination host can ping your host by reverse rules.

3.2 Verification

After deployment of an environment with VMware DVS plugin all corresponding OSTF tests should pass.

3.3 Troubleshooting

VMware dvs driver consists from two parts: the mechanism driver of neutron and the agent. Thereby the main source of information for troubleshooting is `/var/log/neutron/server.log` and `/var/log/neutron/vmware-dvs-agent-....log`.

Please to be sure in correctness of configuration in the `/etc/neutron/plugin.ini` and `/etc/neutron/plugins/ml2/vmware_dvs-.....ini` It should contain following values:

plugin.ini:

```
[ml2]
mechanism_drivers =openvswitch,l2population,vmware_dvs
[ml2_vmware]
vsphere_login=<vsphere_user>
vsphere_hostname=<vsphere_ip>
vsphere_password=<vsphere_password>
```

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.DVSPFirewallDriver

[ml2_vmware]
vsphere_login=<vsphere_user>
network_maps=physnet2:<dvSwitch>
vsphere_hostname=<vsphere_ip>
vsphere_password=<vsphere_password>
```

Sure all neutron-dvs-agent should be launched on corresponded nodes. On controllers — under corosync and on compute-vmware — via init script.

Neutron-dvs-agents must be in active state with cluster host name:

```
root@node-1:~# neutron agent-list -c agent_type -c alive -c host
```

agent_type	alive	host
------------	-------	------

```
agent_type | alive | host |
```

...

Also in case of trouble would be useful to check the connectivity between controller nodes and vCenter.