05 2015

Brocade VCS Plugin Deployment Guide

**In** **Red Hat OpenStack environment**

Copyright © 2012 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, the B-wing symbol, BigIron, DCX, Fabric OS, FastIron, NetIron, SAN Health, ServerIron, and TurboIron are registered trademarks, and Brocade Assurance, Brocade NET Health, Brocade One, CloudPlex, MLX, VCS, VDX, and When the Mission Is Critical, the Network Is Brocade are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned are or may be trademarks or service marks of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. shall have no liability or responsibility to any person or entity with respect to any loss, cost, liability, or damages arising from the information contained in this book or the computer programs that accompany it.

The product described by this document may contain “open source” software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit <http://www.brocade.com/support/oscd>.

# Preface

This document is a deployment guide for implementing a Brocade VCS Plugin, including the key features and options supported NOS device. It is written for technology decision-makers, architects, systems engineers, NOC engineers and other experts responsible for network upgrades and migration.

This document provides step-by-step examples to prepare, perform, and verify the deployment of a Brocade VCS Plugin. It is assumed that the reader is familiar with establishing console access and entering commands using the Brocade CLI.

## Overview

This document describes a Red Hat OpenStack 6.0 installation via PackStack with OpenvSwitch plugin. Once complete, Neutron can be reconfigured to use the Brocade ML2 and SVI driver for managing both virtual and physical networking infrastructure through OpenStack API.

Brocade ML2 and SVI plugin helps to configure L2 and L3 Networking on the underlying Ethernet fabrics from OpenStack Neutron Service.

This document provides an overview of Brocade’s AMPP based ML2 plugin and SVI L3 routing solutions.

This guide has been tested using RHEL 7.0.

## Document History

**Date Version Description**

2015-05-07 1.0 Initial Release

# About Brocade

Brocade® (NASDAQ: BRCD) networking solutions help the world’s leading organizations transition smoothly to a world where applications and information reside anywhere. This vision is designed to deliver key business benefits such as unmatched simplicity, non-stop networking, application optimization, and investment protection.

Innovative Ethernet and storage networking solutions for datacenter, campus, and service provider networks help reduce complexity and cost while enabling virtualization and cloud computing to increase business agility.

To help ensure a complete solution, Brocade partners with world-class IT companies and provides comprehensive education, support, and professional services offerings. ([www.brocade.com](http://www.brocade.com/))

# Glossary

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| VCS | Virtual Cluster and Switching |
| ML2 | Modular Layer 2 |
| NC | Netconf Client |
| NOS | Network Operating System |
| AMPP | Automatic Migration of Port Profiles |
| RHEL | Red Hat Enterprise Linux |
| SVI | Switched virtual interface |

# Topology Overview

VDX 1 – 10.25.225.199

7

VM1

VM2

OVS 1

Eth1thth2

VM3

VM4

OVS 2

VM5

VM6

eth2

VM7

VM8

LB 4

LB3

Red Network

Green Network

Controller

Compute-1

10.24.41.110

Compute-2

10.24.42.66

Eth1

Brocade VCS Fabric

OVS 3

OVS 4

VDX 1 – 10.25.225.200

199/0/12

199/0/13

VM1

# Hardware

* VCS devices running in Logical Chassis Mode

|  |  |
| --- | --- |
| **Platform** | **Description** |
| VDX6710(Carme) | Firmware running v4.0.x,v4.1.x |
| VDX6720(Callisto) | Firmware running v4.0.x,v4.1.x |
| VDX6730(F-Callisto) | Firmware running v4.0.x,v4.1.x |
| VDX8770(Mercury) | Firmware running v4.0.x,v4.1.x,v5.0 |
| VDX6740(Castor) | Firmware running v4.0.x,v4.1.x,v5.0 |
| VDX6740T(Castor-T) | Firmware running v4.0.x,v4.1.x,v5.0 |

# Brocade VCS configuration

Brocade VDX/VCS switches should be running NOS 4.0.x or above with logical chassis mode enabled for configuration distribution across all fabric nodes. See the Brocade NOS administrators guide for additional information. Any ports connected to OpenStack controller, compute and network nodes should be configured as port-profile ports.

**To configure Port-profile-port**

sw0# configure terminal

sw0(config)# interface TenGigabitEthernet 199/0/12

sw0(conf-if-te-199/0/12)# port-profile-port

**To view Port-profile-port configuration**

sw0# show running-config interface TenGigabitEthernet 199/0/12

interface TenGigabitEthernet 199/0/12

port-profile-port

no shutdown

# Server Interface Configuration:

For Tenant networks, edit the configuration of the physical interface connected to the VCS fabric. It should be configured with no IP address and in promiscuous mode. All nodes should have a similar configuration.

NAME=eth1

BOOTPROTO=static

ONBOOT=yes

TYPE=Ethernet

One method to configure the interface for promiscuous mode during boot, is to create /sbin/ifup-local with the following content

#!/bin/bash

if [[ "$1" == "eth1" ]]

then

/sbin/ifconfig $1 promisc

RC=$?

fi

Set executable bit. This script will run during boot right after network interfaces are brought online.

# chmod +x /sbin/ifup-local

# /etc/init.d/network restart

# Install the software repos

Red Hat Enterprise Linux OpenStack Platform requires that each system in the OpenStack environment is running Red Hat Enterprise Linux Server and that all systems be signed up to receive updates from the Customer Portal Subscription Management using Subscription Manager.

Below steps in this procedure must be executed while logged in to the account of the root user on the system being registered.

#subscription-manager register

# subscription-manager list --available

# subscription-manager attach --pool=<pool ID from previous output>

# yum repolist

# subscription-manager repos --enable=rhel-7-server-rpms

# subscription-manager repos --enable=rhel-7-server-openstack-6.0-rpms

# subscription-manager repos --enable=rhel-7-server-rh-common-rpms

# subscription-manager repos --enable=rhel-7-server-optional-rpms

# subscription-manager repos --enable=rhel-7-server-openstack-6.0-installer-rpms

# subscription-manager repos --enable=rhel-server-rhscl-7-rpms

# yum -y update

# reboot

# Installation using PackStack

Below procedure will walk through the process of deploying OpenStack on multi node (one controller and two compute nodes).

Begin by deploying OpenStack as documented in the RHEL OpenStack Platform –

Deploying OpenStack: Proof-of-Concept Environment (PackStack) guides at <https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Deploying_OpenStack_Proof_of_Concept_Environments/index.html> and install PackStack.

yum install -y openstack-packstack

Followed PackStack interactive method to deploy controller and compute nodes. Additional compute nodes can be specified as mentioned below (‘novacompute-hosts’ field).

# packstack

Installer will be installed using the following configuration:

==============================================================

ssh-public-key: /root/.ssh/id\_rsa.pub

default-password:

mariadb-install: y

os-glance-install: y

os-cinder-install: y

os-nova-install: y

os-neutron-install: y

os-horizon-install: y

os-swift-install: y

os-ceilometer-install: y

os-heat-install: n

os-client-install: y

ntp-servers:

nagios-install: y

exclude-servers:

os-debug-mode: n

os-controller-host: 10.24.41.110

os-compute-hosts: 10.24.41.110,10.24.42.66,10.24.42.68

os-network-hosts: 10.24.41.110

os-vmware: n

unsupported: n

use-epel: n

additional-repo:

rh-username:

rhn-satellite-server:

amqp-backend: rabbitmq

amqp-host: 10.24.41.110

amqp-enable-ssl: n

amqp-enable-auth: n

mariadb-host: 10.24.41.110

mariadb-pw: \*\*\*\*\*\*\*\*

keystone-db-passwd: \*\*\*\*\*\*\*\*

keystone-admin-passwd: \*\*\*\*\*\*\*\*

keystone-demo-passwd: \*\*\*\*\*\*\*\*

glance-db-passwd: \*\*\*\*\*\*\*\*

glance-ks-passwd: \*\*\*\*\*\*\*\*

glance-backend: file

cinder-db-passwd: \*\*\*\*\*\*\*\*

cinder-ks-passwd: \*\*\*\*\*\*\*\*

cinder-backend: lvm

cinder-volumes-create: y

cinder-volumes-size: 20G

nova-db-passwd: \*\*\*\*\*\*\*\*

nova-ks-passwd: \*\*\*\*\*\*\*\*

novasched-cpu-allocation-ratio:16.0

novasched-ram-allocation-ratio:1.5

novacompute-migrate-protocol: tcp

os-neutron-ks-password: \*\*\*\*\*\*\*\*

os-neutron-db-password: \*\*\*\*\*\*\*\*

os-neutron-l3-ext-bridge: br-ex

os-neutron-metadata-pw: \*\*\*\*\*\*\*\*

os-neutron-lbaas-install: n

os-neutron-metering-agent-install:n

neutron-fwaas: n

os-neutron-ml2-type-drivers: vlan

os-neutron-ml2-tenant-network-types:vlan

os-neutron-ml2-mechanism-drivers:openvswitch

os-neutron-ml2-flat-networks: \*

os-neutron-ml2-vlan-ranges: physnet1

os-neutron-ml2-tunnel-id-ranges:

os-neutron-ml2-vxlan-group:

os-neutron-ml2-vni-ranges: 10:100

os-neutron-l2-agent: openvswitch

os-neutron-ovs-bridge-mappings:physnet1:br-eth1

os-neutron-ovs-bridge-interfaces:br-eth1:eth1

os-neutron-ovs-tunnel-if:

os-horizon-ssl: n

os-swift-ks-passwd: \*\*\*\*\*\*\*\*

os-swift-storages:

os-swift-storage-zones: 1

os-swift-storage-replicas: 1

os-swift-storage-fstype: ext4

os-swift-storage-size: 2G

provision-demo: y

provision-tempest: n

provision-tempest-user:

provision-tempest-user-passwd: \*\*\*\*\*\*\*\*

provision-demo-floatrange: 172.24.4.224/28

provision-cirros-url: http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86\_64-disk.img

provision-tempest-repo-uri: https://github.com/openstack/tempest.git

provision-tempest-repo-revision:master

provision-all-in-one-ovs-bridge:n

ceilometer-ks-passwd: \*\*\*\*\*\*\*\*

mongodb-host: 10.24.41.110

redis-host: 10.24.41.110

redis-port: 6379

nagios-passwd: \*\*\*\*\*\*\*\*

Proceed with the configuration listed above? (yes|no): yes

# 

# Brocade Neutron plugins

## Ml2 driver description

Brocade NOS plugin for OpenStack Neutron Service implements the Neutron 2 API to manage L2 network on NOS devices.

Brocade NOS Plugin will work with Juno release of OpenStack and extend the following Neutron API’s

#### CREATE\_NETWORK –

On receiving a create\_network call, Brocade Plugin would create a port-profile on the NOS device using the following syntax.

Syntax of the Profile created : openstack-profile-<vlan>

Note:

* vlan number is allocated by the Neutron service based on the range provided in the configuration. Red Network is carried as VLAN 100 and Green Network is carried as VLAN 101 on the VDX box
* Brocade ML2 driver doesn’t support multiple physical network so external network creation will fail
* Ml2 driver supports vlan provider network

#### DELETE\_NETWORK –

On receiving a delete\_network, Brocade Plugin would delete the port-profile on the NOS device corresponding to that Network.

#### CREATE\_PORT –

On receiving a create\_port, Brocade Plugin would create a mac association with port-profile (corresponding to the VLAN) on the NOS device using the mac address of the VM

Note : Openvswitch driver is used for vif port bingind so create\_port operation is dependant on openvswitch

#### DELETE\_PORT –

On receiving a delete\_port, Brocade Plugin would

Delete the mac association with port-profile(corresponding to the VLAN) on the NOS device using the mac address of the VM.

Note : Openvswitch driver is used to remove vif port bingind so delete\_port operation is dependant on openvswitch

## L3 driver description

#### Create\_router, Update\_router

Create\_router and update\_router methods will set the admin\_state state as down

1. add\_interface\_to\_router

Openstack subnet\_id is provided in the request parameters. Using this API, Brocade Plugin will create ve interface and assign gateway ip of the subnet to the VCS device

1. Remove\_interface\_from\_router

Using this API, Brocade Plugin will remove ve interface along with assigned gateway ip of the subnet from VCS device

1. Delete\_router

This method is used to clean up the NETCONF, db connection and any caches.

## Deploy VCS Ml2 driver

On the controller, install the netconf client needed to communicate with the VCS fabric

# git clone https://code.grnet.gr/git/ncclient

# cd ncclient && python setup.py install

Install the Brocade ML2 VCS plugin from the repository on controller nodes.

# yum install –y openstack-neutron-ml2

Edit /etc/neutron/plugins/ml2/ml2\_conf.ini

[ml2]

tenant\_network\_types = vlan

type\_drivers = vlan

mechanism\_drivers = openvswitch,brocade

[ml2\_type\_vlan]

network\_vlan\_ranges = physnet1:100:200

[securitygroup]

enable\_security\_group = True

[database]

connection = mysql://neutron:password@10.24.41.110/neutron

[ovs]

local\_ip = 10.24.41.110

bridge\_mappings = physnet1:br-eth1

integration\_bridge = br-int

[ml2\_brocade]

username = admin

password = password

address = 10.25.225.199

ostype=NOS

physical\_networks = physnet1

osversion = autodetect

Edit /etc/neutron/plugins/ml2/ml2\_conf\_brocade.ini file

[ml2\_brocade]

username = admin

password = password

address = 10.25.225.199

ostype = NOS

physical\_networks = physnet1

osversion = autodetect

Edit /etc/neutron/neutron.conf

core\_plugin = ml2

Make sure the correct network interface is configured to bridges. If interface name is renamed add the renamed interface to the bridge.

#ovs-vsctl add-br br-eth1

#ovs-vsctl add-port br-eth1 eth1

#ovs-vsctl show

941f36d3-1dff-44ec-a930-517bf4083c8d

Bridge br-int

Port "int-br-eth1"

Interface "int-br-eth1"

Port br-int

Interface br-int

type: internal

Bridge "br-eth1"

Port "br-eth1"

Interface "br-eth1"

type: internal

Port "phy-br-eth1"

Interface "phy-br-eth1"

Port "eth1"

Interface "eth1"

Bridge br-ex

Port br-ex

Interface br-ex

type: internal

ovs\_version: "2.0.0"

# 

# SVI - L3 Networking driver.

This section describes how SVI feature can be leveraged to provide internetworking between networks configured using OpenStack.

Edit /etc/neutron/neutron.conf

service\_plugins = neutron.services.l3\_router.brocade.l3\_router\_plugin.BrocadeSVIPlugin

A new field has been added to the existing fields in brocade.ini file.

Add the below configuration in both /etc/neutron/plugins/ml2/ml2\_conf.ini and /etc/neutron/plugins/ml2/ml2\_conf\_brocade.ini

rbridge\_id = <rbridge id of vcs device>

This field indicates the Rbridge on which the SVI interfaces would get created.

# Testing things out – VCS ML2 driver

Source the keystone rc file that was installed into root’s home directory to obtain credentials and use the CLI or Horizon to create networks, and launch new virtual machine instances.

Within the VCS fabric, check that new port-profiles are created for every tenant network that is created.

sw0# show port-profile

port-profile default

ppid 0

vlan-profile

switchport

switchport mode trunk

switchport trunk allowed vlan all

switchport trunk native-vlan 1

port-profile openstack-profile-100

ppid 1

vlan-profile

switchport

switchport mode trunk

switchport trunk allowed vlan add 100

port-profile openstack-profile-101

ppid 2

vlan-profile

switchport

switchport mode trunk

switchport trunk allowed vlan add 101

As new instances are launched, they should be tied to the port-profile corresponding to the network they belong to. Any instances on the same network should be able to communicate with each other.

sw0# show port-profile status

Port-Profile PPID Activated Associated MAC Interface

openstack-profile-100 1 Yes fa16.3e1b.95d0 None

fa16.3e64.fce8 Gi 2/0/28

fa16.3e85.5b2f Gi 2/0/28

fa16.3ea6.3741 Gi 2/0/5

fa16.3ecd.bfc1 Gi 2/0/5

fa16.3eeb.87f7 Gi 2/0/28

openstack-profile-101 2 Yes fa16.3e2c.0baf None

# Testing things out – SVI L3 driver

Use the CLI or Horizon to create router, add interface to the routers.

Configuring the SVI (Ve interface) IP address to be same as the Network’s Gateway IP address would enable us to provide internetworking between the L3 networks.

Red Network is carried as VLAN 100 and Green Network is carried as VLAN 101 on the VDX box (This is achieved using brocade ml2 driver Integration with OpenStack).

VDX Plugin would now configure the router interface IP address as the SVI (Ve) IP address on the switch.

Ve 100 is configured with 6.6.6.1 and Ve 101 is configured with 5.5.5.1

This would allow VM1 (6.6.6.2) on compute-1 to communicate with VM7(5.5.5.7) on compute-2