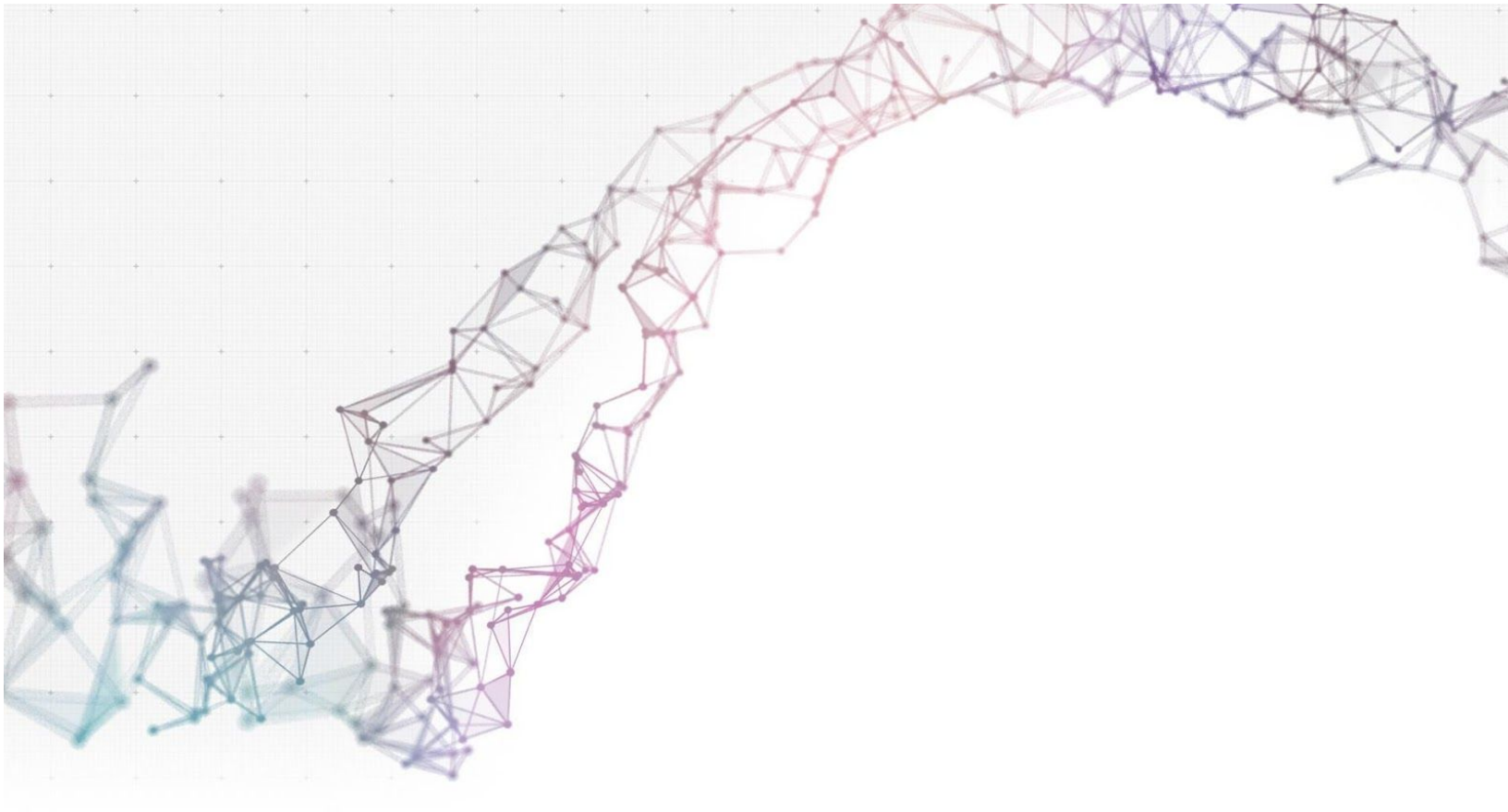

INSTALLATION RUNBOOK FOR Juniper vSRX

Application Type:	**VNF (virtual firewall)**
vSRX Version:	**15.1X49-D30.3**
MOS Version:	**8.0**
OpenStack Version:	**Liberty**

1 Introduction

This document is to serve as a detailed Deployment Guide for Juniper vSRX deployed with Mirantis OpenStack (MOS). This document describes the reference architecture, installation steps for validated vSRX with Mirantis OpenStack, limitations and testing procedures.

1.1 Target Audience

This guide is designed for OpenStack Administrators who are deploying Juniper vSRX with Mirantis OpenStack.

2 Application Overview

2.1 Juniper vSRX Overview

vSRX is a virtualized next generation firewall that provides security and networking services at the perimeter or edge in virtualized private or public cloud environments. vSRX runs as a virtual machine (VM) on a standard x86 server.

vSRX next generation firewall provides visibility into applications, users and also threat visibility, protection, enforcement and control. vSRX is built on Junos OS and delivers networking and security features similar to those available on SRX Series Services Gateways for the branch. Some of the key benefits of vSRX in virtualized private or public cloud multitenant environments include:

- Stateful firewall protection at the tenant edge
- Faster deployment of virtual firewalls
- Full routing, VPN, and networking capabilities
- Centralized and local management

3 Joint Reference Architecture

The reference architecture used for this runbook is as follow:

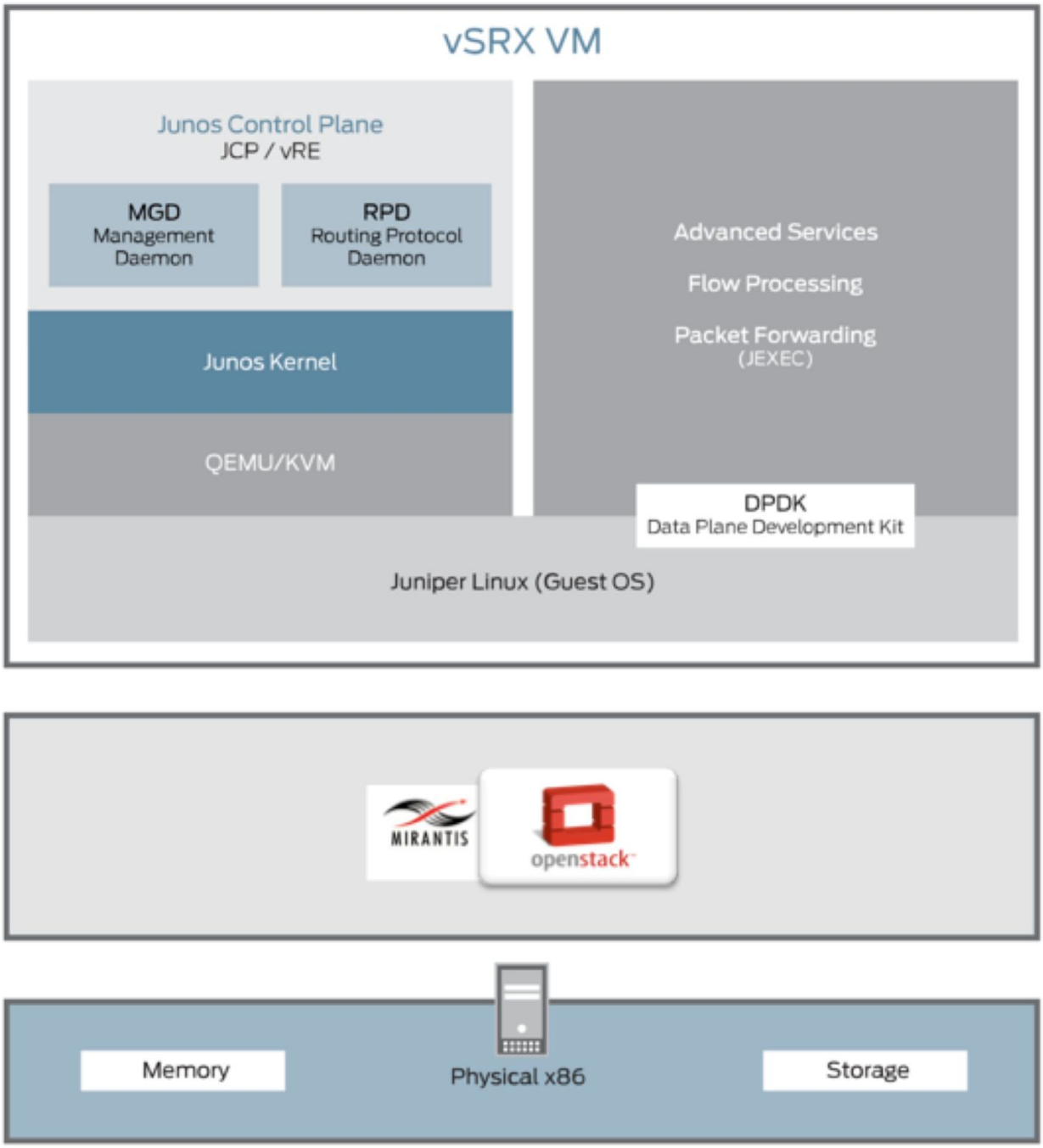

Figure 1 vSRX joint architecture with Mirantis OpenStack

Below are the high level steps to bring up vSRX on Mirantis OpenStack:

1. **Creation of OpenStack environment**
   Download and install MOS 8.0
2. **Preinstall checks**
   Check versions of OS, grub, kernel, KVM, libvirt, virsh connectivity test to QEMU, IXGBE
3. **Setup internals**

Create and Deploy Mirantis OpenStack environment

4. **Bring up vSRX**

4 Physical & Logical Network Topology

The setup and testing in this runbook will be based on the logical and physical topology outlined in *Figure 2*.

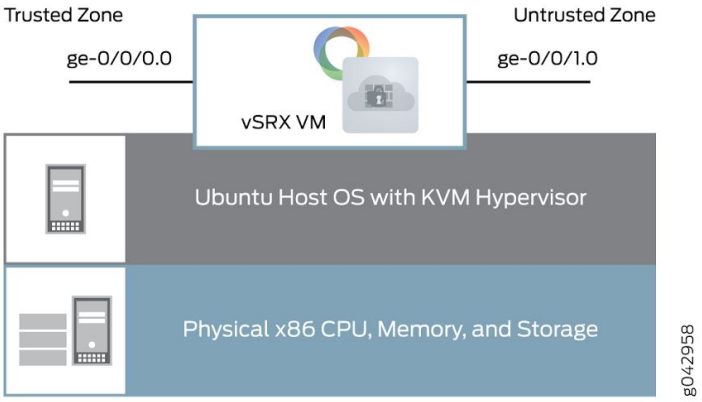

Figure 2 vSRX Network Topology with MOS

5 Installation & Configuration

5.1 MOS environment preparation

Compute node must be a bare metal, because Juniper vSRX VNF does not work with nested virtualization.

5.1.1 MOS configuration

OS	Mode	Hypervisor	Networking model	Storage	Additional services
Ubuntu	HA	KVM	Neutron+VLAN/VXLAN	any	any

5.1.2 Creation of OpenStack environment

1. Download 8.0 MOS ISO from Mirantis website.
2. Boot the Fuel Master node.
3. Create a new OpenStack environment in the following configuration:

a. KVM as hypervisor
b. Neutron+VLAN or VXLAN as networking option

5.1.3 Health Check Results

OpenStack Testing Framework (OSTF also known as Health Checks) provides ability to verify the cluster operability in the post-deployment stage. You can find more information on Health Checks in the official Mirantis OpenStack documentation.

5.2 vSRX installation steps

1. Download the vSRX KVM appliance image from the Juniper website.

2. Create a Glance image:
```
# glance image-create --container-format bare --property hw_disk_bus=ide  --disk-format
qcow2 --progress --file media-vsrx-vmdisk-15.1X49-D30.3.qcow2 --name vsrx
```

3. Create a flavor for vSRX according to its Software environment requirements:

```
# nova flavor-create vSRX auto 8000 40 8
```

4. Create required networks and subnets with necessary properties (in our case we've used 3 networks - management, trusted and untrusted):

```
# neutron net-create mgmt_net
# neutron subnet-create mgmt_net 10.168.2.0/24 --name mgmt_subnet --enable-dhcp
# neutron net-create trusted_net
# neutron subnet-create trusted_net 10.20.20.0/24 --name trusted_subnet --enable-dhcp
--allocation-pool start=10.20.20.3,end=10.20.20.254 --gateway 10.20.20.2
# neutron net-create untrusted_net
# neutron subnet-create untrusted_net 10.20.21.0/24 --name untrusted_subnet
--enable-dhcp --allocation-pool start=10.20.21.3,end=10.20.21.254 --gateway 10.20.21.2
```

5. Run vSRX instance:

```
# nova boot --image vsrx --flavor vSRX --nic net-id=<mgmt_net_id> --nic
net-id=<trusted_net_id> --nic net-id=<untrusted_net_id> vsrx-1
```

6. In Horizon, open the console and wait until vSRX instance is installed:

```
Starting install...
<output omitted>
The machine id is empty.
Cleaning up ...
Thu Aug 27 12:06:22 UTC 2015
```

```
Aug 27 12:06:22 init: exec_command: /usr/sbin/dhcpd (PID 1422) started
Aug 27 12:06:22 init: dhcp (PID 1422) started
Aug 27 12:06:23 init: exec_command: /usr/sbin/pppd (PID 1428) started

Amnesiac (ttyd0)

Login:
```

7. Log into vSRX instance console and verify the vSRX version is properly installed:

```
login: root
--- JUNOS 15.1X49-D30.3 built 2015-12-17 04:39:24 UTC
root@% cli
root> show version
Model: vSRX
Junos: 15.1X49-D30.3
JUNOS Software Release [15.1X49-D30.3]
```

Verify that the interfaces and packet forwarding engine (PFE) are up and running on vSRX:

```
root> show chassis fpc pic-status
Slot 0   Online       Virtual FPC
  PIC 0  Online       Virtual GE

root> show interfaces terse | match ge-
ge-0/0/0                up    up
ge-0/0/0.0              up    up   inet     172.19.101.187/24
ge-0/0/1                up    up
ge-0/0/1.0              up    up   inet     192.168.113.254/24
```

8. Set up root password and configure the management interface.
Create a root password:

```
root@% cli
root> edit
[edit]
root# set system root-authentication plain-text-password
```

Set the IP address family for the management interface, and enable the DHCP client for this interface:

```
[edit]
root#set interfaces fxp0 unit 0 family inet dhcp-client
Commit
```

Now you can connect to the vSRX instance by ssh via management network.

9. For configuration of more vSRX features, please refer to the vSRX Administration Guide for KVM.

5.6 Testing

5.6.1 vSRX

For basic testing, we configure trusted and untrusted networks in the vSRX, and check routing between these networks.

1. Configure interfaces, security zones and create virtual router for the vSRX instance:

```
root@% cli
root> edit
# set interfaces ge-0/0/0 unit 0 family inet address 10.20.20.2/24
# set interfaces ge-0/0/1 unit 0 family inet address 10.20.21.2/24
# set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic
system-services all
# set security zones security-zone untrust interfaces ge-0/0/1.0 host-inbound-traffic
system-services icmp
# set routing-instances CUSTOMER-VR instance-type virtual-router
# set routing-instances CUSTOMER-VR interface ge-0/0/0.0
# set routing-instances CUSTOMER-VR interface ge-0/0/1.0
# commit
```

2. Configure Neutron ports for the vSRX instance:

```
# neutron port-update <trusted_net_port_id> --allowed_address_pairs list=true type=dict
ip_address=10.20.21.0/24,ip_address=10.20.20.2
# neutron port-update <untrusted_net_port_id> --allowed_address_pairs list=true
type=dict ip_address=10.20.20.0/24 ip_address=10.20.21.2
```

3. Launch two VMs with interfaces in trusted and untrusted networks respectively:

```
# nova boot --image TestVM --flavor m1.tiny --nic net-id=<trusted_net_id> vm-trusted
# nova boot --image TestVM --flavor m1.tiny --nic net-id=<untrusted_net_id>
vm-untrusted
```

4. Try to *ping* vSRX interfaces from two VMs:
    a. From vm-trusted - ge-0/0/0.0 (10.20.20.2), *ping* should work
    b. From vm-untrusted - ge-0/0/1.0 (10.20.21.2), *ping* should work
    c. Try to ping vm-trusted from vm-untrusted, *ping* should not work
    d. Try to ping vm-untrusted from vm-trusted, *ping* should work.