IMPLEMENTATION GUIDE

Gi-LAN Solution Implementation Installation Guide

Intel Corporation
Datacenter Networking Solution Group

Authors

Eduardo Castro
Solution Software Engineer, Intel Corporation

Tarek Radi
Lead Technical Program Manager, Intel Corporation

1.0 Introduction

As Communications Service Providers move to a software-defined networking/ network function virtualization (SDN/NFV) world, they find themselves needing to enable several use cases. This solution implementation document focuses on the Gateway-Internet LAN (Gi-LAN) use case and presents how to build a single host virtualized Gi-LAN architecture that can serve mobile customer traffic using both local content as well as Internet accessibility, while enabling key traffic roaming policies.

This document describes the configuration and integration of several virtual network functions (VNFs) from various third-party suppliers on top of a Red Hat Enterprise Linux*-based network function virtualization infrastructure (NFVI), running Red Hat OpenStack* Platform 7. The VNFs used in this solution are given in Table 1.

Table 1. Virtual functions of the solution.

REQUESTED VIRTUAL FUNCTION	SUPPLIER	PRODUCT/VERSION
Switching	Open vSwitch Community	Open vSwitch* 2.3.2
Routing	Brocade	5600 vRouter* build: 3.2.1R6
Domain Name System (DNS)		
Policy and Charging Rules Function (PCRF), Policy and Charging Enforcement Function (PCEF), and Traffic Detection Function (TDF)	Sandvine	Policy Traffic Switch Virtual Series* 7.00.01 Service Delivery Engine Virtual Series* 7.10 Subscriber Policy Broker Virtual Series* 6.50
Management user interface for PCRF and PCEF/TDF		Control Center* 6.90.02
Firewall	F5 Networks	BIG-IP* Advanced Firewall Manager 11.6.0
Distributed Denial of Service (DdoS)		BIG-IP Carrier-Grade NAT 11.6.0
Carrier-Grade Network Address Translation (CGNAT)		
Transparent Proxy and Caching	The Apache Software Foundation	Apache Traffic Server* 6.0.0
Content Delivery Network		
Origin Web Server		Apache HTTP Server* 2.4.10
Outbound Anti-Spam	Cisco Systems	Snort* 2.9.6.0

Table of Contents

The primary audiences for this document are architects and engineers planning to implement their own virtualized Gi-LAN architectures. While this document discusses the solution in detail, it is by no means a large-scale solution that can handle thousands of users. For example, the topology here uses only one host server. Readers should use this document as proof that such a Gi-LAN use case is possible in SDN/NFV. They can follow the steps documented here to build their own proof-of-concept topology and scale it.

The intent of this document is to help customers who are interested in implementing this specific use case in an SDN/NFV world. It is important to note that the details contained herein are just an example of one way of enabling this use case for a customer. Intel does not aim to promote or recommend any specific hardware, software, or supplier mentioned in this document. In addition, Intel does not aim to tie customers to any specific software and hardware stack. We recommend that you use this document only as a guide to enable this use case in an SDN/NFV world.

For an overview of this Gi-LAN solution, including the hardware and software components used, please refer to the Gi-LAN Solution Implementation Summary.

Link to the Solution Implementation Summary: https://networkbuilders.intel.com/network-technologies/solution-blueprints

2.0 Installation Guide

This section contains instructions for installing and configuring the software stack on a single-server hardware platform based on Intel® Xeon® processor family. This solution is built with a set of VNF elements that form the Gi-LAN functional chain, and are deployed on a variety of networks as presented in Table 2.

Table 2. Networks used in the solution.

	NETWORK	IP
1	Input	172.16.88.0/24
2	After Router	10.10.10.0/24
3	After Network Policy Control	10.10.10.0/24
4	After Firewall	12.12.12.0/24
5	After Snort	12.12.12.0/24
6	After DNS	16.16.16.0/24
7	After HTTP Caching	16.16.16.0/24
8	After CDN	15.15.15.0/24
10	Services	21.21.21.0/24
11	Management	172.16.77.0/24
12	Output	10.250.100.0/24

2.1 Prerequisites

Please note that the instructions for installing Red Hat Enterprise Linux 7.1 are not within the scope of this document; however, this section contains some remarks that the user should follow during operating system installation or configuration.

For a successful installation, please make sure you do the following:

- Create a RAID 0 virtual disk from all the physical disks.
- Create custom partitioning as presented in Table 3.

Table 3. Solution partitioning schema.

PARTITION	SIZE
Biosboot	2 MB
/boot	5 GB
/swap	Double the size of physical memory
/ (root partition)	Remaining space

To start, execute the following steps.

1. Install Red Hat Enterprise Linux 7.1, and then set the following:
   - Hostname: `/etc/hostname`
   - Hosts: `/etc/host`
   - Disable Security Enhanced Linux: `/etc/selinux/conf`
2. Disable and stop the firewall.

```
# systemctl disable firewalld
# systemctl stop firewalld
```

3. Set the networks interfaces (they should follow the pattern `/etc/sysconfig/network-scripts/ifcfg-em1`).

```
TYPE=Ethernet
BOOTPROTO=static
IPADDR=172.16.77.2
NETMASK=255.255.255.0
GATEWAY=172.16.77.1
DNS1=10.248.2.1
DNS2=10.2.71.6
DNS3=10.19.1.4
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=em1
UUID=633b582e-7696-4e43-ad1d-ac53bf074250
DEVICE=em1
ONBOOT=yes
NM_CONTROLLED=no
```

4. Disable and stop the NetworkManager, and then enable/start the network and bring up the respective network interface.

```
# systemctl disable NetworkManager
# systemctl stop NetworkManager
# systemctl start network
# systemctl enable network
# ifup em1
```

2.2 Red Hat OpenStack Platform 7

1. Set the proxy on /etc/yum.conf and update proxy information.

```
proxy=http://proxy-us.intel.com:911
```

2. Export the proxy configuration.

```
# export http_proxy=http://proxy-us.intel.com:911
# export https_proxy=http://proxy-us.intel.com:911
```

3. Subscribe the Red Hat and Red Hat OpenStack Platform 7.

```
# subscription-manager register
# subscription-manager subscribe --auto
# subscription-manager list --consumed
```

4. Clear the initially set up repositories and enable the appropriate ones.

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-rh-common-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-7.0-rpms
```

5. Install the PackStack installer.

```
# yum install openstack-packstack
```

6. Create an answer file. The answer file used in this setup is presented in Appendix A: PackStack Answer File.

7. Run PackStack.

```
# packstack —-answer-file=/home/myanswerfile.txt
```

2.3 Carrier-Grade NAT

1. Make sure that the image of F5 BIG-IP CGNAT is configured to have an IDE hard disk and e1000 interface cards. Note that “F5” is the image name.

```
# . /root/keystonerc_admin
# glance image-update --property hw_disk_bus=ide F5
# glance image-update --property hw_vif_model=e1000 F5
```

2. Create a flavor in OpenStack and call it CGNAT with the specs:

   - Virtual CPUs: 4
   - RAM: 6 GB
   - Hard Disk (HD): 150 GB

3. Create an instance from the F5 image and call it CGNAT. Network connections are in this order:
   - Management
   - Services
   - After_HTTP_Caching
   - Output
   - Log in to the CGNAT console.
   - User: root
   - Password: default

4. Edit /PLATFORM.

```
platform=Z100 family=0xC0000000 host=Z100 systype=0x71
```

5. In the command line area, execute:

```
# reboot
# config
```

   - Enter the management IP address 172.16.77.208, and the netmask 255.255.255.0.
   - Default route is 172.16.77.209 (Dell server IP address on the management network).

6. In a web browser, type https:// 172.16.77.208 or its equivalent in your deployment.
   - Check “I understand the risks”.
   - Add exceptions.
   - Confirm security exception.

7. Log in with admin/admin credentials.

8. Set up the utility.
   - Click Next.
   - Click Activate.
   - Enter the base registration key (should include LTM CGN modules enabled).
   - Select activation method: Manual.
   - Click Next.
   - Copy the Dossier.
   - Click “Click here to access F5 Licensing Server”.
   - Paste the Dossier.
   - Click Next.
   - Check “I have read and agree to the terms of this license”.
   - Click Next.
   - Copy the content display inside the display area.
   - Return to the tab of the F5 graphical user interface (GUI).
   - Paste the copied content.
   - Click Next.

9. Enable modules.
   - Enable CGNAT.
   - Click Next and then OK.
   - Click Next.
   - Set both the root and admin passwords [for example "123456"].
   - Log in again with admin and the password you set in the previous step.
   - Click Finished.
10. Network→VLAN
    - Click Create.
    - Name: Vlan_services
    - Interface: 1.1
    - Tagging: untagged
    - Click Add.
    - Click Finished.
    - Click Create.
    - Name: Vlan_internal
    - Interface: 1.2
    - Tagging: untagged
    - Click Add.
    - Click Finished.
    - Click Create.
    - Name: Vlan_external
    - Interface: 1.3
    - Tagging: untagged
    - Click Add.
    - Click Finished.
11. Network→Self IPs
    - Click Create.
    - Name: Services_selfIP
    - IP Address: 21.21.21.16
    - Netmask: 255.255.255.0
    - Vlan_services
    - Click Finished.
    - Click Create.
    - Name: Internal_selfIP
    - IP Address: 16.16.16.3
    - Netmask: 255.255.255.0
    - Vlan_internal
    - Click Finished.
    - Click Create.
    - Name: External_selfIP
    - IP Address: 10.250.100.148
    - Netmask: 255.255.255.0
    - Vlan_external
    - Click Finished.
12. Carrier Grade NAT→LSN Pools
    - Click Create.
    - Name: pool_external
    - Egress: Enabled on
    - Vlan_output
    - Address/Prefix Length: 105.203.255.70/32 (this should be the IP address Vlan_external in case you don't have a real IP)
13. Local Traffic→Virtual servers
    - Name: virtualservernat
    - Type: Forwarding (IP)
    - Source Address: 0.0.0.0/0
    - Destination Address: 0.0.0.0/0
    - Service port: *
    - Protocol: All Protocols
    - VLAN and Tunnel Traffic: Enabled on
    - Vlan_internal
    - Source Address Translation: LSN
    - LSN Pool: pool_external
    - Click Finished.
14. Network→Routes
    - Name: default route
    - Destination: 0.0.0.0
    - Netmask 0.0.0.0
    - Resource: Use Gateway
    - Gateway Address: IP Address: 10.250.100.145
    - Click Finished.

2.4 Transparent Proxy - Caching

1. Create a flavor in OpenStack and call it "TP" with the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 80 GB
2. Create a CentOS* instance (from the CentOS image, which includes development tools like gcc, g++, etc.) called TransparentProxy with the previous flavor. The instance should have interfaces with these networks:
   - Management
   - Services
   - After_DNS
   - After_HTTP_Caching
3. Install extra tools.

```
# yum install wget bzip2
# yum install gcc gcc-c++ pkgconfig pcre-
devel tcl-devel expat-devel openssl-devel
# yum install libcap libcap-devel hwloc
hwloc-devel ncurses-devel libcurl-devel
# yum install autoconf automake libtool
# yum install ebtables
# yum install bridge-utils
```

4. Set the network interfaces—edit the following files in the /etc/sysconfig/network-scripts/ directory.

   • /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=172.16.77.53
NETMASK=255.255.255.0
```

   • /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=21.21.21.28
NETMASK=255.255.255.0
```

   • /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
BRIDGE="br0"
```

   • /etc/sysconfig/network-scripts/ifcfg-eth3

```
DEVICE="eth3"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
BRIDGE="br0"
```

   • /etc/sysconfig/network-scripts/ifcfg-br0

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=none
ONBOOT=yes
STP=off
IPADDR=16.16.16.8
NETMASK=255.255.255.0
GATEWAY=16.16.16.3
DNS1=8.8.8.8
```

5. Restart the network service, and bring up all the network interfaces.

```
# systemctl restart network

# ifup <interface>
```

6. Install the Apache traffic server. For more information, follow:
   https://docs.trafficserver.apache.org/en/6.1.x/admin-guide/installation/index.en.html.

```
# wget http://archive.apache.org/dist/trafficserver/trafficserver-6.0.0.tar.bz2
# tar xvfj trafficserver-6.0.0.tar.bz2
# cd trafficserver-6.0.0
# ./configure --prefix=/opt/ts —enable-posix-cap —enable-tproxy=force
# make
# make check
# sudo make install
# opt/ts/bin/traffic_server -R 1
```

7. Modify ../ts/etc/trafficserver/records.config as shown below. For details, see:
   https://docs.trafficserver.apache.org/en/4.2.x/reference/configuration/records.config.en.html

```
CONFIG proxy.config.http.server_ports STRING 8080:tr-full
CONFIG proxy.config.url_remap.remap_required INT 0
CONFIG proxy.config.cluster.ethernet_interface STRING br0
```

8. Apply the changes.

```
# traffic_line -x
```

9. Enable the IP table rules.

```
# ebtables -t broute -F
# ebtables -t broute -A BROUTING -p IPv4 --ip-proto tcp --ip-dport 80 -j redirect --redirect-target DROP
# ebtables -t broute -A BROUTING -p IPv4 --ip-proto tcp --ip-sport 80 -j redirect --redirect-target DROP
# iptables -t filter --flush FORWARD
# iptables -t filter --flush INPUT
# iptables -t mangle -A PREROUTING -i eth2 -p tcp -m tcp --dport 80 -j TPROXY --on-ip 0.0.0.0 --on-port 8080 --tproxy-mark 1/1
# iptables -t mangle -A PREROUTING -i eth3 -p tcp -m tcp --sport 80 -j MARK --set-mark 1/1
# ip rule add fwmark 1/1 table 1
# ip route add local 0.0.0.0/0 dev lo table 1
# echo 1 > /proc/sys/net/ipv4/ip_forward
# echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
# echo 0 > /proc/sys/net/ipv4/conf/br0/rp_filter
# echo 0 > /proc/sys/net/ipv4/conf/eth2/rp_filter
# echo 0 > /proc/sys/net/ipv4/conf/eth3/rp_filter
```

10. Start the Traffic Server.

```
# trafficserver start
```

11. You can browse the cache content with the following command.

```
# /opt/ts/bin/traffic_logstats | less
```

2.5 Domain Name Service

1. Create a flavor in OpenStack and call it “Brocade” with the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 20 GB
2. Create a volume in OpenStack and BrocadeVol:
   - HD: 10 GB
3. Create an instance “BrocadeInstall” from the Brocade ISO image and call it “BrocadeISO” with flavor Brocade. Network connections:
   - Management
4. Shut off the “BrocadeInstall” instance.
5. Attach the volume to the instance.
6. Turn on the instance again.
7. When you go to the console of the Brocade 5600 vRouter, type "install image".
8. Proceed with the installation and confirm erasing the HD along with modifying the GRUB. Keep the username and password “vyatta” and “vyatta”.
9. When the installation finishes, turn off the instance.
10. Detach the volume from the instance created.
11. From the volume properties, upload the volume as an image.
12. Delete the instance.
13. Once you upload the new Image “Brocade” you can instantiate the image as a Nova instance to be used as a router.
14. Create an instance “DNS” from the Brocade image previously created with flavor Brocade. Network connections are in this order:
    - Management
    - Services
    - After_Snort
    - After_DNS
15. Log in with “vyatta”/“vyatta”.
16. Type the following commands in the Brocade command line interface. For more information follow: http://www.brocade.com/content/html/en/vrouter5600/42r1/vrouter-42r1-laninterface/GUID-E8AFD5DC-CF6E-47C3-BD07-B9CDB34405DF.html

```
configure
set interfaces dataplane dp0s3 address 172.16.77.207/24
set interfaces dataplane dp0s4 address 21.21.21.15/24
set interfaces dataplane dp0s5 address 12.12.12.10/24
set interfaces dataplane dp0s6 address 16.16.16.2/24
set protocols static route 0.0.0.0/0 next-hop 16.16.16.3
set protocols static route 10.0.0.0/8 next-hop 12.12.12.3
set service https
set service dns forwarding cache-size 1000
set service dns forwarding listen-on dp0s5
set service dns forwarding listen-on dp0s6
set service dns forwarding name-server 8.8.8.8
set service dns forwarding name-server 8.8.4.4
commit
save
```

2.6 Outbound Anti-Spam (Snort)

1. Create a flavor in OpenStack and call it “antispam” with the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 50 GB

1. Create an instance “AntiSpam” from the CentOS (http://docs.openstack.org/image-guide/obtain-images.html) with flavor “antispam”. Network connections in this order:
   - Management
   - Services
   - After_Firewall
   - After_Snort
2. Install some extra tools.

```
# yum install net-tools
# yum install wget make gcc flex bison zlib zlib-devel libpcap libpcap-devel pcre pcre-devel tcpdump gcc-c++ libdnet libdnet-devel
```

3. Disable networking and execute.

```
$ sudo su
```

4. Edit /etc/rc.local and add the following.

```
ifconfig eth2 up
ifconfig eth3 up
```

5. Edit the following files in the /etc/network/interfaces directory.

- /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=172.16.77.74
NETMASK=255.255.255.0
```

- /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=21.21.21.45
NETMASK=255.255.255.0
```

- /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
```

- /etc/sysconfig/network-scripts/ifcfg-eth3

```
DEVICE="eth3"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
```

6. Install Snort.

```
# yum install https://www.snort.org/
downloads/snort/daq-2.0.6-1.centos7.x86_64.
rpm
# yum install https://www.snort.org/
downloads/snort/snort-2.9.8.3-1.centos7.
x86_64.rpm
# wget https://www.snort.org/rules/
community
#tar -xvfz community.tar.gz -C /etc/snort/
rules
```

Alternatively, you can execute the following steps.

```
# wget https://www.snort.org/downloads/
snort/snort-2.9.8.3.tar.gz
# tar xvfz snort-2.9.8.3.tar.gz
# cd snort-2.9.8.3
# ./configure --enable-sourcefire && make
&& sudo make install
# ldconfig
```

Create a new snort.conf file with DAQ variables configured for AFPacket. For details, follow: https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/013/original/Snort_IPS_using_DAQ_AFPacket.pdf

```
config daq: afpacket
config data_mode: inline
config policy_mode: inline
reject icmp any any -> <16.16.16.3> any
```

7. Start Snort.

```
# snort --daq afpacket –Q –c /root/snort.
conf –l /root/log –i eth2:eth3
```

2.7 Firewall and DDoS

Assuming that the image properties have already been updated as described in the CGNAT configuration, we proceed as follows:

1. Create a flavor in OpenStack and call it “Firewall” with the following specs:
   - Virtual CPUs: 6
   - RAM: 12 GB
   - HD: 150 GB
2. Create an instance “Firewall” from the F5 BIG-IP image with flavor “Firewall”. Network connections in this order:
   - Management
   - Services
   - After_Network_Policy_Control
   - After_Firewall

3. Log in to the Firewall console with root/default.
4. Edit /PLATFORM.

```
platform=Z100 family=0xC0000000 host=Z100
systype=0x71
```

5. Execute:

```
# reboot
# config
```

- Enter the management IP address 172.16.77.201, and the netmask 255.255.255.0.
- Default route 172.16.77.209 (Dell server IP address on the management network).

6. In a web browser type https://172.16.77.201 or its equivalent in your deployment.
   - Select "I understand the risks".
   - Add exceptions.
   - Confirm security exception.
7. Log in with admin/admin.
8. Setup Utility.
   - Click Next.
   - Click Activate.
   - Enter the base registration key (should include LTM CGN modules enabled).
   - Activation method: Manual
   - Click Next.
   - Copy the Dossier.
   - Click "click here to access F5 Licensing Server".
   - Paste the Dossier.
   - Click Next.
   - Check "I have read and agree to the terms of this license".
   - Click Next.
   - Copy the content display inside the display area.
   - Return to the tab of the F5 GUI.
   - Paste the copied content.
   - Click Next.
9. Enable Modules
   - Enable LTM, AFM, PEM, ASM.
   - Click Next and then OK.
   - Click Next.
   - Set both the root and admin passwords (for example, "123456").
   - Log in again as admin using the password you have set previously.
   - Click Finished.
10. Network→VLAN→Create
    - Name: Vlan_services
    - Interface: 1.1
    - Tagging: untagged
    - Click Add.
    - Click Finished.
11. Network→VLAN→Create
    - Name: Vlan_internal
    - Interface: 1.2
    - Tagging: untagged
    - Click Add.
    - Click Finished.
12. Network→VLAN→Create
    - Name: Vlan_external
    - Interface: 1.3
    - Tagging: untagged
    - Click Add.
    - Click Finished.
13. Network→Self IPs→Create
    - Name: Services_selfIP
    - IP Address: 21.21.21.9
    - Netmask: 255.255.255.0
    - Vlan_services
    - Click Finished.
14. Network→Self IPs→Create
    - Name: Internal_selfIP
    - IP Address: 10.10.10.5
    - Netmask: 255.255.255.0
    - Vlan_internal
    - Click Finished.
15. Network→Self IPs→Create
    - Name: External_selfIP
    - IP Address: 12.12.12.3
    - Netmask: 255.255.255.0
    - Vlan_external
    - Click Finished.
16. Local Traffic→Virtual Servers→Create
    - Name: NAT
    - Type: Forwarding (IP)
    - Source Address: 0.0.0.0/0
    - Destination Address: 0.0.0.0/0
    - Service Port: *
    - Protocol: * All Protocols
    - VLAN and Tunnel Traffic: Enabled on
      - vlan_internal
      - vlan_external
    - Source Address Translation: Auto Map
    - Click Finished.
17. Security→Address Lists→Create
    - Name: youtube_IPs
    - Add the IP addresses from this website.
    - Click Finished.

18. youtubeSecurity→Policies
    - Name: PoC_Policy
    - Click Finished.
19. Open PoC_Policy→Add
    - Name: Allow_any
    - Order: Last
    - Action: Accept
    - Click Finished.
20. Open PoC_Policy→Add
    - Name: Drop_youtube
    - Order: First
    - Protocol: TCP 6
    - Destination: Address/region
    - Address List
    - /Common/youtube_IPs
    - Port
    - Port
    - 80
    - Action: Drop
    - Click Finished.
21. Security→DoS Protection
    - Open: dos
    - Application Security: Mark it
    - Protocol Security (DNS): Mark it
    - Protocol Security (SIP): Mark it
    - Network Security: Mark it
    - Update
22. Network→Routes
    - Name: Default_GW
    - Destination: 0.0.0.0
    - Netmask 0.0.0.0
    - Resource: Use Gateway
    - Gateway Address: IP Address: 12.12.12.10
    - Click Finished.
    - Name: Subscriber
    - Destination: 10.0.0.0
    - Netmask 255.0.0.0
    - Resource: Use Gateway
    - Gateway Address: IP Address: 10.10.10.2
    - Click Finished.

2.8 Brocade Input

1. Create a flavor in OpenStack and call it “Brocade” with the following specs:
    - Virtual CPUs: 2
    - RAM: 4 GB
    - HD: 20 GB
2. Create a volume in OpenStack and BrocadeVol1.
    - HD: 10 GB
3. Create an instance “BrocadeInstall” from the Brocade ISO image and call it “BrocadeISO” with flavor Brocade. Network connections:
    - Management
4. Shut off the “BrocadeInstall” instance.
5. Attach the volume to the instance.
6. Turn on the instance again.
7. When you go to the console of the Brocade 5600 vRouter, type "install image".
8. Proceed with the installation and confirm erasing the HD along with modifying the GRUB. Keep the username and password “vyatta” and “vyatta”.
9. When the installation finishes, turn off the instance.
10. Detach the volume from the instance created.
11. From the volume properties, upload the volume as an image.
12. Delete the instance.
13. Once you upload the new Image “Brocade1” you can instantiate the image as a Nova instance to be used as a router.
14. Create an instance “DNS” from the “Brocade1” image previously created with flavor Brocade. Network connections in this order:
    - Management
    - Services
    - Input
    - After_Router
15. Log in with “vyatta” “vyatta”.
16. Type the following commands. For more information, follow: http://www.brocade.com/content/html/en/vrouter5600/42r1/vrouter-42r1-laninterface/GUID-E8AFD5DC-CF6E-47C3-BD07-B9CDB34405DF.html.

```
configure
set interfaces dataplane dp0s3 address
172.16.77.210/24
set interfaces dataplane dp0s4 address
21.21.21.2/24
set interfaces dataplane dp0s5 address
172.16.88.133/24
set interfaces dataplane dp0s6 address
10.10.10.2/24
set protocols static route 0.0.0.0/0 next-
hop 10.10.10.5
set service nat source rule 100 outbound-
interface dp0s6
set service nat source rule 100 source
address 10.0.0.0/8
set service nat source rule 100
translation address masquerade
set service https
commit
save
```

17. Please note that in this deployment this router does not have Dynamic Host Configuration Protocol (DHCP) server enabled on its input interface "dp0s5", because the Gateway GPRS Support Node (GGSN) has a fixed IP address and forwards the traffic directly to this interface. In lab environments, however, you would need to enable DHCP on the router so that test clients would have IP addresses in the range the PoC expects.

```
set service dhcp-server shared-network-name server subnet 10.20.30.0/24 start 4 stop 10.20.30.100
set service dhcp-server shared-network-name server subnet 10.20.30.0/24 dns-server 12.12.12.10
set service dhcp-server shared-network-name server subnet 10.20.30.0/24 default-router 10.20.30.2
```

Note: 12.12.12.10 is the IP of Brocade DNS; 10.20.30.2 is the IP of the router having DHCP server; 10.20.30.0/24 is the input network.

2.9 Policy Traffic Switch (PTS)

1. Make sure that the image of the PTS is configured to have IDE hard disk and e1000 interface cards. PTS is the image name.

```
$ sudo su
# . /root/heystonerc_admin
# glance image-update --property hw_disk_bus=ide PTS
# glance image-update --property hw_vif_model=e1000 PTS
```

2. Create a flavor in OpenStack and call it "PTS" with the following specs:

- Virtual CPUs: 2
- RAM: 2 GB
- HD: 15 GB

3. Create an instance from the PTS image and call it "PTS". Network connections in this order:

- Management
- Services
- After_Router
- After_Network_Policy_Control

4. Log in to the PTS console with credentials provided by Sandvine.

5. Edit `/etc/rc.conf.local`. 172.16.77.211 is the IP address of the server on the management network.

```
ifconfig_em0="inet 172.16.77.211/24"
ifconfig_em1="inet 21.21.21.3/24"
defaultrouter="172.16.77.209"
```

6. Execute.

```
# svcli
PTS> configure
PTS# set config service spb servers 21.21.21.4
PTS# set config cluster sub-name pts-virtual-series
PTS# set config cluster stat-name pts-virtual-series
PTS# commit
PTS# save config
```

2.10 DHCP

1. Create an instance "BrocadeInstall" from the Brocade ISO image and call it "BrocadeISO" with flavor Brocade. Network connections in this order:

- Management
- Services
- Input

2. Log in with "vyatta"/"vyatta".

3. Configure the DHCP with the Brocade console.

```
configure
set interfaces dataplane dp0s3 address dhcp
set service ssh
set service https http-redirect disable
set interfaces dataplane dp0s4 address 21.21.21.46/24
set interfaces dataplane dp0s5 address 172.16.88.16/24
set interfaces dataplane dp0s3 address 172.16.77.75/24
set protocols static route 0.0.0.0/0 next-hop 172.16.88.10 distance 1
set service dhcp-server disabled false
set service dhcp-server shared-network-name LAN subnet 172.16.88.0/24 default-router 172.16.88.10
set service dhcp-server shared-network-name LAN subnet 172.16.88.0/24 dns-server 12.12.12.8
set service dhcp-server shared-network-name LAN subnet 172.16.88.0/24 domain-name internal-network
set service dhcp-server shared-network-name LAN subnet 172.16.88.0/24 lease 86400
set service dhcp-server shared-network-name LAN subnet 172.16.88.0/24 start 172.16.88.100 stop 172.16.88.200
commit
save
```

2.11 Service Delivery Engine (SDE)

1. Make sure that the image of the SDE is configured to have IDE hard disk and e1000 interface cards. SDE is the image name.

```
$ sudo su
# . /root/heystonerc_admin
# glance image-update --property hw_
disk_bus=ide SDE
# glance image-update --property hw_vif_
model=e1000 SDE
```

2. Create a flavor in OpenStack and call it “SDE” with the following specs:
   - Virtual CPUs: 1
   - RAM: 4 GB
   - HD: 20 GB
3. Create an instance from the SDE image and call it “SDE”. Network connections in this order:
   - Management
   - Services
4. Log in to the SDE console with credentials provided by Sandvine.
5. Edit /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
IPADDR=172.16.77.197
NETMASK=255.255.255.0
GATEWAY=172.16.77.209
DNS1=8.8.8.8
TYPE="Ethernet"
```

6. Edit /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="static"
ONBOOT="yes"
IPADDR=21.21.21.5
NETMASK=255.255.255.0
TYPE="Ethernet"
```

7. Execute:

```
# iptables -t nat -A PREROUTING -i eth0
-p udp --dport 67 -j REDIRECT --to-port
3128
# iptables -t nat -A PREROUTING -i eth0
-p udp --dport 68 -j REDIRECT --to-port
3128
# iptables -t nat -A PREROUTING -i eth0
-p udp --dport 546 -j REDIRECT --to-port
3129
# iptables -t nat -A PREROUTING -i eth0
-p udp --dport 547 -j REDIRECT --to-port
3129
# svcli
SDE> configure
SDE# set config service spb servers
21.21.21.4
SDE# commit
SDE# save config
```

2.12 Subscriber Policy Broker (SPB)

1. Make sure that the image of the SPB is configured to have IDE hard disk and e1000 interface cards. SPB is the image name.

```
$ sudo su
# . /root/heystonerc_admin
# glance image-update --property hw_
disk_bus=ide SPB
# glance image-update --property hw_vif_
model=e1000 SPB
```

2. Create a flavor in OpenStack and call it SPB with the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 15 GB
3. Create an instance from the SPB image and call it “SPB”. Network connections in the order:
   - Management
   - Services
4. Log in to the SDE console with credentials provided by Sandvine.
5. Edit /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
IPADDR=172.16.77.196
NETMASK=255.255.255.0
GATEWAY=172.16.77.209
DNS1=8.8.8.8
TYPE="Ethernet"
```

6. Edit /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="static"
ONBOOT="yes"
IPADDR=21.21.21.4
NETMASK=255.255.255.0
TYPE="Ethernet"
```

2.13 Control Center

1. Create a flavor in OpenStack and call it “Control Center” with the specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 50 GB
2. Create an instance “Control Center” from a fresh Windows* Server 2012 image with flavor “Control Center”. Network connections in the order:
   - Management
   - Services
3. Set the management interface to IP address 172.16.77.198/24 and gateway 172.16.77.209
4. Set the services interface to IP address 21.21.21.6/24
5. Install the Sandvine Control Center software.
6. Open the Sandvine Control Center application, and then click OK.
7. Create a new connection:
   - Name: sv-spb
   - Host: 21.21.21.4
   - Port: 8443
8. Click Finish, and then click Close.
9. Log in with the sv-spb root/123456.
10. New Datahome
    - Display name: sv-spb
    - SPB cluster name: spb-virtual-series
    - SPB host name: 21.21.21.4
    - Click Finish.
    - Click Next.
11. In order to apply subscriber mapping that will be used for PCRF, we need to have radius messaging sent from the GGSN to the SDE through the management network.
12. Solutions→Configuration→Add
    - Subscriber Mapping, and then click OK.
    - Click Next.
    - Click Next.
    - Radius, and then click Next.
    - Select the SDE and the PTS, move them to the right, and then click Next.
    - SDE IP: 21.21.21.5, and then click Next.
    - Subnets file should contain:
      - 0.0.0.0/0
      - 10.0.0.0/8
      - ::/0
    - Click Next.
    - Radius interfaces 172.16.77.197 with port number 1813 and shared secret “nfv” (as configured with GGSNs).
    - Another source sends RADIUS messages to the SDE, and the SDE will not acknowledge messages.
    - Click Next.
    - Use advanced mode to manually edit the configuration.
    - Replace every ‘User-Name’ with ‘Calling-Station-Id’.
    - Done.
    - Deploy the changes from the command center.

Note: The previous subscriber mapping only works if there is a GGSN. For local lab testing, we need to apply DHCP subscriber mapping; this is coupled with enabling DHCP on the Brocade input. In order to do DHCP mapping, proceed to step 13. If not, jump to step 14.

13. Solutions→Configuration→Add
    - Subscriber Mapping, and then click OK.
    - Click Next.
    - Click Next.
    - DHCPv4, and then click Next.
    - Select the SDE and the PTS, move them to the right, and then click Next.
    - SDE IP: 21.21.21.5 and the click Next.
    - Click Next.
    - Subnets file should contain:
      - 0.0.0.0/0
      - 10.0.0.0/8
      - ::/0
    - Click Next.
    - Delete DHCPv6.
    - PTS will rewrite each packet with PTS as source and SDE as destination.
    - Use port 3128.
    - Add 10.0.0.0/8.
    - Click Finish.
- Deploy the changes.

2.14 Policy and Charging Rules Function

In the following example, we define three subscription categories (Gold, Silver, and Bronze). Users who have a Gold subscription have a maximum download speed of 24 Mbps. Silver and Bronze users can download with a maximum speed of 4 Mbps and 384 Kbps, respectively.

1. Open the Control Center Console.
2. Policy
   - Pts-virtual-series
   - PowerEdit
3. Add to policy the sizes of the packages that would be controlled by the PolicyGroup. In order to create the script to map the subscriber flows use the reference of the Sandvine documentation. (http://documents.mx/download/link/sde-sandscript-reference-guide-r640-a02pdf)
4. To assign a subscription package to users, follow the steps below.
   - First you have to get the user's name. From the Solutions-Monitoring-Audits logs, get the names of the mapped users.

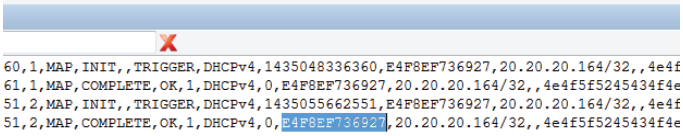


Figure 1. Retrieving the user name from Solutions-Monitoring-Audit log.

   - From Operations-SPB-Browser, click subscriber.
   - Click “subscriber name”, and then write the subscriber's name in the white box to ensure that you have the correct name. Then click submit.
   - Click set attribute value.
   - In the “attribute name” white box, write “package”.
   - In the “attribute value” white box, write the package name (gold, silver, or bronze).
   - Click submit.

2.15 Content Delivery Network

1. Create a flavor in OpenStack and call it “CDN” with the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 80 GB
2. Create a CentOS instance (from the CentOS image, which include development tools like gcc, g++, etc) called ContentDeliveryNetwork with the previous flavor. The instance should have interfaces with these networks:
   - Management
   - Services
   - After_Network_Policy_Control
   - After_CDN
3. Install extra tools.

```
# yum install wget bzip2
# yum install gcc gcc-c++ pkgconfig pcre-devel tcl-devel expat-devel openssl-devel
# yum install libcap libcap-devel hwloc hwloc-devel ncurses-devel libcurl-devel
# yum install autoconf automake libtool
```

4. Set network interfaces by editing the following files in /etc/sysconfig/network-scripts/.
   - /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=172.16.77.70
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=21.21.21.41
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=10.10.10.9
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth3

```
DEVICE="eth3"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=15.15.15.13
NETMASK=255.255.255.0
```

5. Restart network service, and if-up all the network interfaces.

```
# systemctl restart network
```

6. Install Apache traffic server. For more information, follow: (https://docs.trafficserver.apache.org/en/6.1.x/admin-guide/installation/index.en.html).

```
# wget http://archive.apache.org/dist/trafficserver/trafficserver-6.0.0.tar.bz2
# tar xvfj trafficserver-6.0.0.tar.bz2
```

```
# cd trafficserver-6.0.0
# ./configure --prefix=/opt/ats
# make
# make check
# sudo make install
# cd /opt/ats
# bin/traffic_server -R 1
```

7. Add the line below to /opt/ts/etc/trafficserver/remap.config to SET configuration for Origin servers. For more information, follow: https://docs.trafficserver.apache.org/en/6.1.x/admin-guide/files/remap.config.en.html.

```
map http://10.10.10.9 http:// 10.10.10.9 / \
   @plugin=/opt/ts/libexec/trafficserver/
   balancer.so @pparam=--
   policy=roundrobin @pparam=15.15.15.10
   @pparam=15.15.15.11
```

Make sure that line below in /opt/ts/etc/trafficserver/records.config looks like:

```
CONFIG proxy.config.http.server_ports
STRING 80
```

8. Start CDN.

```
# bin/trafficserver start
```

2.16 Origin Server1

1. Create a flavor in OpenStack and call it “OriginServer” with the specs:
   - Virtual CPUs: 1
   - RAM: 2 GB
   - HD: 50 GB
2. Create a CentOS instance “OriginServer1” with flavor “OriginServer”. Network connections in the order:
   - Management
   - Services
   - After_CDN
3. Install the httpd service:

```
# yum install httpd
# systemctl enable httpd
```

4. Set the hostname:

```
# hostnamectl  set-hostname webserver1
```

5. Set Network interfaces by editing the following files in /etc/sysconfig/network-scripts/.
   - /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=172.16.77.49
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=21.21.21.22
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=15.15.15.10
NETMASK=255.255.255.0
```

6. Restart the network.

```
# systemctl restart network
```

7. Edit /var/www/html/index.html.

```
<html>
Server1
</html>
```

8. Start the httpd service.

```
# systemctl start httpd
```

2.17 Origin Server2

1. Create a flavor in OpenStack and call it “OriginServer” with the following specs:
   - Virtual CPUs: 1
   - RAM: 2 GB
   - HD: 50 GB
2. Create an instance “OriginServer2” with flavor “OriginServer”. Network connections are in this order:
   - Management
   - Services
   - After_CDN
3. Install the httpd service.

```
# yum install httpd
# systemctl enable httpd
```

4. Set hostname.

```
# hostnamectl set-hostname webserver2
```

5. Set network interfaces by editing the following files in /etc/sysconfig/network-scripts/.
   - /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=172.16.77.50
NETMASK=255.255.255.0
```

   - /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="none"
```

```
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=21.21.21.23
NETMASK=255.255.255.0
```

- /etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
BOOTPROTO="none"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
IPV6INIT="no"
IPADDR=15.15.15.11
NETMASK=255.255.255.0
```

6. Restart the network service.

```
# systemctl restart network
```

7. Edit /var/www/html/index.html.

```
<html>
Server2
</html>
```

8. Start the httpd service.

```
# systemctl start httpd
```

3.0 Test Input

To test the whole setup, test that the virtual machine (VM), CentOS or Windows, is connected to the input of the setup. The "Test Input" is the name used in this document to refer to this virtual name. Test Input represents the user whose traffic has to pass through the NFV setup before going to the Internet. Test Input VM is configured at the beginning. After that, it is used in all use cases of this section.

3.1 Configuring Test Input

1. Create a new flavor in OpenStack and call it "CentOS". Use the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 50 GB
2. Create an instance "Attacker" from the CentOS image with flavor "CentOS". Network connections in the order:
   - Management
   - Services
   - Input
3. Open the console of Test input and disable networking by editing /etc/sysconfig/network-scripts/.

```
DEVICE=eth0
IPADDR=172.16.77.149
NETMASK=255.255.255.0
DEVICE=eth1
iface eth1 inet static
IPADDR=21.21.21.18
DEVICE=eth2
IPADDR=172.16.88.134
NETMASK=255.255.255.0
GATEWAY=16.16.16.3
DNS1=12.12.12.10
```

4. Restart the network service.

```
# systemctl restart network.
```

5. Now the test input should be able to ping any node in the setup.

3.2 Accessing the Internet

1. Open the console of the Test Input (which represents a user). Then open a web browser. On the web browser type the hostname of any webpage. If the test input is able to browse the Internet, this means that the:
   - The Brocade input router is able to route the user's traffic to the firewall through the network policy control.
   - The firewall applied its default rule by accepting user's traffic.
   - DNS has done the address name translation of the website.
   - CGNAT has transformed the private IP address of the user (Test Input) to a public one.

3.3 Blocking Websites

1. On the console of the Test Input VM, open the browser, and then type "www.youtube.com". Two rules apply:
   - Default rule: This rule accepts all user's traffic unless there is another rule.
   - Block youtube rule: This rule blocks www.youtube.com.
2. You will find that the webpage will not load. This means that the firewall applied its second rule by blocking "www.youtube.com" or its equivalent according to your configuration.
3. To view the rules enforced by the firewall, log in to the Firewall GUI. On a web browser, type https:// 172.16.77.201 or its equivalent in your deployment. Log in with admin/ admin.
4. Security→Network Firewall
   - Active rules
   - Now you can see the applied rules that you have configured (in our case "Drop_youtube" and "Allow_any").
5. To view statistics of the applied rules:
   - Log in to the firewall GUI as done in step 3.
   - Move the mouse over "Security".
   - Click "Overview".
   - You can now see how many times each rule is applied.
   - On the Network→Top rules drop-down menu, click "Line Chart".
   - You can see the following figure.

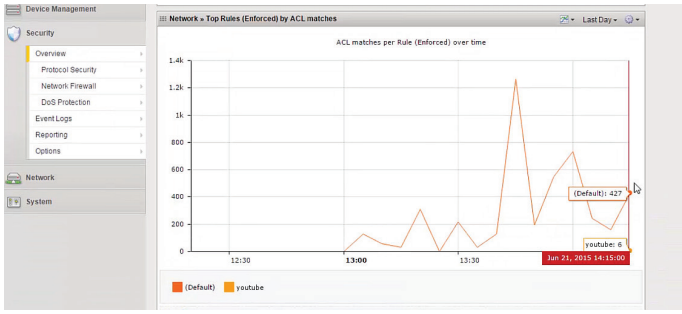


Figure 2. Viewing statistics of applied rules in the firewall GUI.

3.4 Viewing Cached Content

1. To test the HTTP caching functionality:

- Open the Test Input console, and then open an HTTP website for the first time.
- Clear the browser cached content on the test input VM.
- Reload the page.
- You will find that the content will load faster than the first time because some of the content is cached at HTTP transparent proxy caching VM.

Note: Cached content on the browser of the Test Input VM was cleared to make sure that the browser cache did not take part in increasing the speed of page loading the second time.

2. To view the cached content from the transparent proxy VM, open the console of transparent proxy VM:

```
# cd /opt/ts/bin/
./traffic_logstats | less
```

Now you can see the number of cache hits and misses for each HTTP website.

3.5 Viewing DNS Statistics

1. Open the browser of the Test Input VM and access different websites. Note that for every new website that you type on the browser, a new DNS request goes to the DNS VM of our setup. Note that the DNS VM in our setup is configured to do DNS caching and forwarding.

2. To view the DNS statistics, open the Brocade DNS console, and then type:

```
# show dns forwarding statistics
```

Now you can see the total DNS queries inserted to cache, the number of queries forwarded, and the number of queries served locally and other statistics.

3. To view DNS Statistics from the Brocade DNS Dashboard, open the dashboard by typing https:// 172.16.77.207/ on the host machine's browser (the server's browser).

You can see DNS forwarding under Services.

Clicking the arrow beside "DNS forwarding" displays the DNS statistics.

3.6 Carrier-Grade NAT Statistics

1. The CGANT transforms the private IP of users (in our case: test input) to public ones.
2. From the CGNAT GUI it is possible to see some statistics of the NATted requests.

- Open the CGNAT web interface by typing "https://172.16.77.208/" on the host machine's browser.
- Move the mouse over "Carrier Grade NAT".
- Move the mouse over "LSN Pools".
- Press Statistics.

3.7 Users' Data Rate Using Router

1. Log in to the Brocade 5600 vRouter's dashboard by typing "https://30.30.30.210" on the host server's browser or its equivalent in your setup.
2. Select any interface; for example, input interface (dp0s4), or the output interface (dp0s5).
3. A new graph with statistics for the interface appears, which shows you the data rate of the ongoing and outgoing traffic on the selected interface.

3.8 User's Traffic Using Network Policy Control

In this section, we show how to use network policy control to monitor the traffic browsed by Test Input.

1. Open the console of the Control Center.
2. You will see connected Sandvine PTS/SPB/SDE under the Operations-Inventory on the left menu as shown in the following figure:

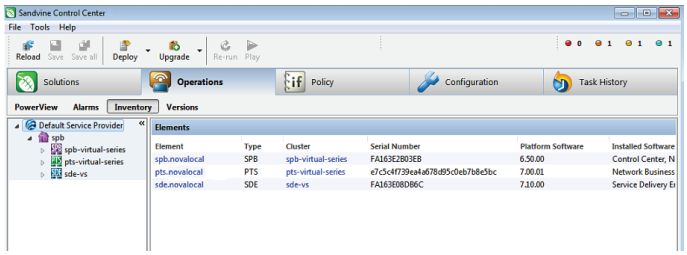


Figure 3. Connected Sandvine PTS/SPB/SDE Virtual Series* visible under the Operations-Inventory view in Sandvine Control Center*.

3. To see the breakdown of the traffic, under pts-virtual-series choose the PTS. You should see the following:

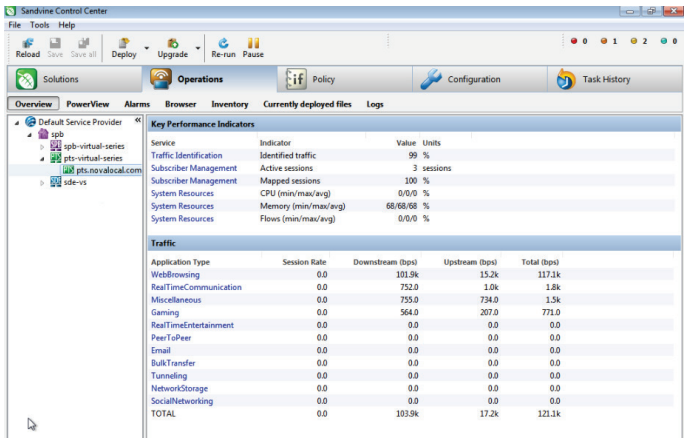


Figure 4. Traffic breakdown under Sandvine PTS Virtual Series* in Sandvine Control Center*.

You can select WebBrowisng or RealTimeCommunication to see more details. You should be able to monitor the websites that the Test Input is currently browsing.

3.9 Balancing between Servers Using CDN

1. From the test input VM open the browser, and then type the CDN's IP (http://10.10.10.9/).
2. You will find that the browser displays the HTML content of one of the two origin servers.
3. Clear the browser's cache.
4. Repeat steps 1 through 3.
5. You will find that the browser balances between displaying the two origin servers' content. This means that the CDN is connected to two origin servers and is serving some users' requests from one server and others from the second server. This prevents the origin server from getting overloaded.
6. Note that we have hosted different content on each origin server to make sure the CDN balances between both servers. However, in a practical deployment both servers should host the same content.

3.10 Preventing Attacks Using DDoS Protection

To make sure the DDoS is functioning properly, we have to connect a new VM (Attacker VM) to the setup as shown in Figure 5. The attacker VM is used to generate a DoS attack to test the DDoS's ability to prevent attacks.

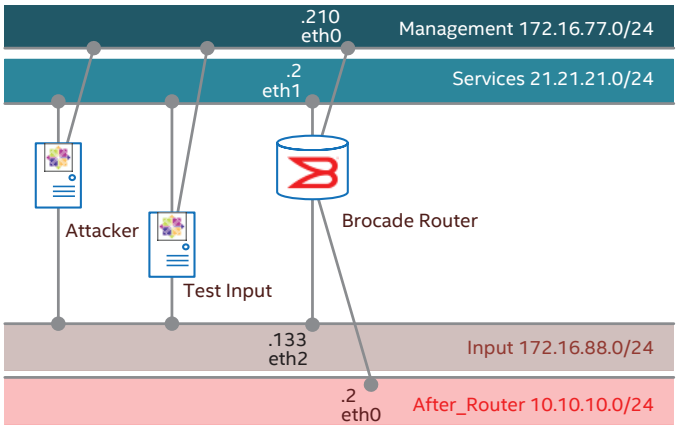


Figure 5. Attacker VM connected to the setup.

The following steps describe how to configure the Attacker VM and how to generate a DoS attack.

1. Use the "CentOS" flavor created before. Recall that it has the following specs:
   - Virtual CPUs: 2
   - RAM: 4 GB
   - HD: 50 GB
2. Create an instance "Attacker" from the CentOS image with flavor "CentOS". Network connections in the order:
   - Management
   - Services
   - Input
3. From the VM, disable networking by editing /etc/sysconfig/network-scripts/.

```
DEVICE=eth0
IPADDR=172.16.77.150
NETMASK=255.255.255.0

DEVICE=eth1
IPADDR=21.21.21.18

DEVICE=eth2
IPADDR=172.16.88.135
NETMASK=255.255.255.0
GATEWAY=16.16.16.3
DNS1=12.12.12.10
```

4. Restart the network service.

```
# systemctl restart network
```

5. Download the NetPerf packet generator.

```
$ sudo yum install netperf
```

6. Use the packet generator to flood the network by generating a lot of requests (DoS attack) as follows.

```
# NetPerf -H 10.250.100.148 –t TCP_STREAM
--m 1024
```

The following steps stop an attack using F5 DDoS protection.

1. Get to know how the DDoS is enabled in the F5 instance.
2. Log in to the F5 GUI. On a web browser, type https://172.16.77.201 or its equivalent in your deployment. Log in with admin/admin.
3. From the left menu, select Security→Overview.
4. From the top menu, select DoS.
5. From the left menu, select Reporting→DoS.
6. You will find the DDoS protection has stopped an attack of severity "5" as shown in the following figure.

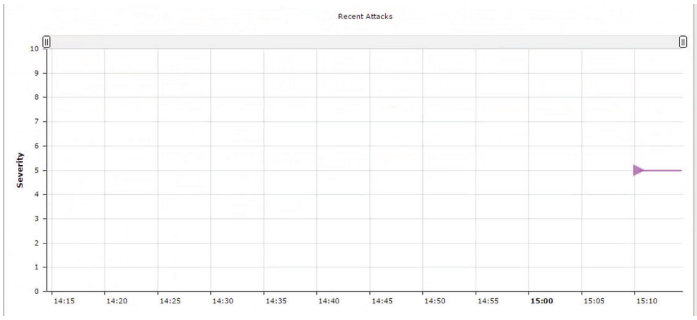


Figure 6. Attack stopped by F5 DDoS protection.

3.11 Enforcing Policy Using PCRF

1. Open the console of the Control Center.
2. First you have to get the user's name. From Solutions-Monitoring-Audits logs, get the name corresponding to the test input IP.

3. From Operations-SPB-Browser, click subscriber.
4. Click subscriber name, and then write the subscriber's name in the white box to ensure that you have the correct name. Then click submit.
5. Click set attribute value.
   - In the “attribute name” white box, write “package”.
   - In the “attribute value” white box, write the package name (gold, silver, or bronze).
6. Click submit.

4.0 Test Cases for Components

4.1 Carrier-Grade NAT

1. The CGNAT is working if the traffic can access the Internet through it.
2. From the CGNAT GUI it is possible to see some statistics of the NATted.
3. Carrier Grade NAT→LSN Pools→Statistics

4.2 Transparent Proxy

1. To view the cached content from the transparent proxy VM, execute the following command.

```
# /opt/ts/bin/traffic_logstats | less
```

   It is also possible to test the content cached by trying to browse an HTTP website for the first time from a VM connected to the input of the transparent, then clear the browser cached content, and then reload the page. The content will load faster than the original time because some of the content is cached.

4.3 Domain Name Service

1. To view the DNS content from the Brocade DNS console, execute the following command.

```
show dns forwarding statistics
```

2. Alternatively, from the Brocade DNS HTML webpage, you can see DNS forwarding under Services by clicking the arrow; the DNS cached hits are displayed.
3. A VM connected to the DNS can also set its DNS name-server to 12.12.12.10; the HTTP traffic on this VM will work as the Brocade DNS acts as a DNS server.

4.4 Outbound Anti-Spam (Snort)

1. The anti-spam reject log is found `in /root/snort.conf`.
2. The CGNAT “16.16.16.3” is blocked in the configuration file.
3. Any VM after the anti-spam will not be able to ping 16.16.16.3.

4.5 Firewall

To view the rules enforced by the firewall:

1. Log in to the Firewall GUI. In a web browser, type https://172.16.77.201 “or its equivalent in your deployment”. Log in with admin/admin.
2. Security→Network Firewall→Active Rules
3. Now you can see the applied rules that you have configured (In our case “Drop_youtube” and “Allow_any”).

To test the firewall functionality, follow the steps below.

1. You can create a CentOS VM before the firewall to represent the user (that is, the VM’s traffic has to pass through the firewall before going to the Internet).
2. On the VM’s browser, type: www.youtube.com
3. The web page is blocked.

4.6 CDN and Origin Servers

1. From a test VM that represents the user, open the browser and type the CDN’s IP (http://10.10.10.9/).
2. The browser will display the HTML content of one of the two origin servers.
3. Clear the browser’s cache.
4. Repeat steps 1 through 3.
5. The browser will balance between displaying the two servers’ content. This means that the CDN has two servers and is serving some users’ requests from one server and others from the second server. This prevents the origin server from getting overloaded. Note that in real-case scenarios both origin servers should host the same content. We made them host different content to validate that the CDN and origin servers are working properly.

4.7 Brocade Router

1. Create a Test VM at the input of the router.
2. If the VM is able to access the Internet, the router properly routed the users’ traffic to the firewall through the network policy control.
3. You can view some traffic statistics from the router’s dashboard.
   - Log in to the Brocade router’s dashboard by typing “https://30.30.30.210” on the host server’s browser or its equivalent in your setup.
   - Select any interface; for example, input interface (dp0s4) or the output interface (dp0s5).
   - You can view the data rate of the ongoing and outgoing traffic on the selected interface as shown below.

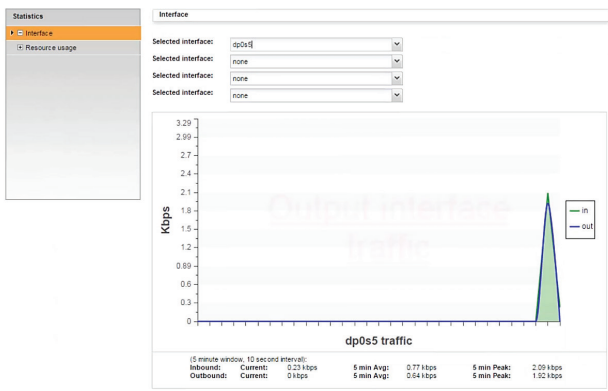


Figure 7. Brocade router statistics.

4.8 Network Policy Control

The following steps show how to use network policy control to monitor the traffic browsed by a VM connected to the Sandvine VNF.

1. Open the console of the Control Center.
2. You should be able to see connected PTS/SPB/SDE under the Operations-Inventory on the left menu.
3. To see the breakdown of the traffic, under pts-virtual-series choose the PTS.
4. You can select WebBrowsing or RealTimeCommunication to see more details. You should be able to monitor the websites that the Test Input is currently browsing.

5.0 Sample Test Scenarios

This section presents example performance test scenarios defined to benchmark the setup benchmarked with iPerf*/ iPerf3* tool. These tests measure the throughput and latency at the specific locations of the VNF chain. For the purpose of benchmarking, several additional VM were created.

5.1 Scenario 1: Performance at the Input Router

The test input is located in the Subscriber (Input) network, and the results are collected at the egress of the input router. To set up the scenario, follow the steps below.

1. Create a CentOS 7-based VM in the Subscriber (Input) network, and call it Test_Subcriber. IP is assigned by DHCP.
2. Create a CentOS-based VM in the After_Router network (after Input router), and call it Test_Router.
3. Install iPerf3 on both VMs.

```
# yum install iperf3
```

4. Start iPerf3 in the Test_Router VM.

```
# iperf3 –s
```

5. Start iPerf3 in the Test_Subcriber.

```
# iperf3 –c <ip_of_Test_Router> –t 60
```

5.2 Scenario 2: Performance at the Firewall

The test input is located in the Subscriber (Input) network, and the results are collected at the egress of the firewall. To set up the scenario, follow the steps below.

1. Create or reuse the VM as described in step 1 of Scenario 1.
2. Create a CentOS 7-based VM in After_Network_Policy_ Control network (after Input router), and call it Test_D.
3. Install iPerf3 on Test_D VM.

```
# yum install iperf3
```

4. Start iperf3 in the Test_D VM.

```
# iperf3 –s
```

5. Start iPerf3 in the Test_Subcriber.

```
# iperf3 –c <ip_of_Test_D> –t 60
```

5.3 Scenario 3: Performance after the Firewall

For this test, CGNAT VM was skipped. The test input is located after the firewall, and the results are collected at the extra router (output router) created in place of CGNAT. To set up the scenario, follow the steps below.

1. Create a CentOS 7-based VM in the After_Snort network, and call it Test_A. Manually assign the addresses: 12.12.12.x/24, and 12.12.12.8 for the gateway.
2. Create a VM in the Output network, and call it Test _Output. Assign a free IP address from the network 10.250.100.0. Default gateway must point to the output router (10.250.100.40).
3. Install iPerf3 on both VMs.

```
# yum install iperf3
```

4. Configure a static route in the DNS router.

```
set protocols static route 0.0.0.0/0 next-hop 16.16.16.19 distance 1
```

5. Start iPerf3 in the Test_Output VM.

```
# iperf3 –s
```

6. Start iPerf3 in the Test_A VM.

```
# iperf3 –c <ip_of_Test_Output> –t 60
```

5.4 Scenario 4: Performance of the Network Chain

This test measures the performance of the network chain—from Subscriber (Input) network to the performance router (Output network). To set up the scenario, follow the steps below.

1. Reuse Test_Subcriber and Test_Output VMs.
2. Start iPerf3 in the Test_Output VM.

```
# iperf3 –s
```

3. Start iPerf3 in the Test_Subscriber VM.

```
# iperf3 –c <ip_of_Test_Output> –t 60
```

5.5 Results

Table 4 presents the average throughput and latency results measured for each scenario over 60 seconds. Note that the latency was measured with the ping tool.

Table 4. Test results.

	AVERAGE THROUGHPUT [GBPS]	AVERAGE LATENCY [MS]
Scenario 1	3.55	1.03
Scenario 2	0.93	1.93
Scenario 3	3.04	2.22
Scenario 4	0.33	7.43

Appendix A: PackStack Answer File

```
[general]
    CONFIG_SSH_KEY=/root/.ssh/id_rsa.pub
    CONFIG_DEFAULT_PASSWORD=<set your
    password here>
    CONFIG_MARIADB_INSTALL=y
    CONFIG_GLANCE_INSTALL=y
    CONFIG_CINDER_INSTALL=y
    CONFIG_MANILA_INSTALL=n
    CONFIG_NOVA_INSTALL=y
    CONFIG_NEUTRON_INSTALL=y
    CONFIG_HORIZON_INSTALL=y
    CONFIG_SWIFT_INSTALL=y
    CONFIG_CEILOMETER_INSTALL=y
    CONFIG_HEAT_INSTALL=n
    CONFIG_SAHARA_INSTALL=n
    CONFIG_TROVE_INSTALL=n
    CONFIG_IRONIC_INSTALL=n
    CONFIG_CLIENT_INSTALL=y
    CONFIG_NTP_SERVERS=
    CONFIG_NAGIOS_INSTALL=y
    EXCLUDE_SERVERS=
    CONFIG_DEBUG_MODE=n
    CONFIG_CONTROLLER_HOST=172.16.77.2
    CONFIG_COMPUTE_HOSTS=172.16.77.2
    CONFIG_NETWORK_HOSTS=172.16.77.2
    CONFIG_VMWARE_BACKEND=n
    CONFIG_UNSUPPORTED=n
    CONFIG_USE_SUBNETS=n
    CONFIG_VCENTER_HOST=
    CONFIG_VCENTER_USER=
    CONFIG_VCENTER_PASSWORD=
    CONFIG_VCENTER_CLUSTER_NAME=
    CONFIG_STORAGE_HOST=172.16.77.2
    CONFIG_SAHARA_HOST=172.16.77.2
    CONFIG_USE_EPEL=n
    CONFIG_REPO=
    CONFIG_ENABLE_RDO_TESTING=n
    CONFIG_RH_USER=
    CONFIG_SATELLITE_URL=
    CONFIG_RH_PW=
    CONFIG_RH_OPTIONAL=y
    CONFIG_RH_PROXY=
    CONFIG_RH_PROXY_PORT=
    CONFIG_RH_PROXY_USER=
    CONFIG_RH_PROXY_PW=
    CONFIG_SATELLITE_USER=
    CONFIG_SATELLITE_PW=
    CONFIG_SATELLITE_AKEY=
    CONFIG_SATELLITE_CACERT=
    CONFIG_SATELLITE_PROFILE=
    CONFIG_SATELLITE_FLAGS=
    CONFIG_SATELLITE_PROXY=
    CONFIG_SATELLITE_PROXY_USER=
    CONFIG_SATELLITE_PROXY_PW=
    CONFIG_SSL_CACERT_FILE=/etc/pki/tls/
    certs/selfcert.crt
    CONFIG_SSL_CACERT_KEY_FILE=/etc/pki/
    tls/private/selfkey.key
    CONFIG_SSL_CERT_DIR=~/packstackca/
    CONFIG_SSL_CACERT_SELFSIGN=y
    CONFIG_SELFSIGN_CACERT_SUBJECT_C=--
    CONFIG_SELFSIGN_CACERT_SUBJECT_
ST=State
    CONFIG_SELFSIGN_CACERT_SUBJECT_L=City
    CONFIG_SELFSIGN_CACERT_SUBJECT_
O=openstack
    CONFIG_SELFSIGN_CACERT_SUBJECT_
OU=packstack
    CONFIG_SELFSIGN_CACERT_SUBJECT_
CN=poc
    CONFIG_SELFSIGN_CACERT_SUBJECT_
MAIL=admin@poc
    CONFIG_AMQP_BACKEND=rabbitmq
    CONFIG_AMQP_HOST=172.16.77.2
    CONFIG_AMQP_ENABLE_SSL=n
    CONFIG_AMQP_ENABLE_AUTH=n
    CONFIG_AMQP_NSS_CERTDB_PW=<set your
password here>
    CONFIG_AMQP_AUTH_USER=amqp_user
    CONFIG_AMQP_AUTH_PASSWORD=<set your
password here>
    CONFIG_MARIADB_HOST=172.16.77.2
    CONFIG_MARIADB_USER=root
    CONFIG_MARIADB_PW=intel
    CONFIG_KEYSTONE_DB_PW=intel
    CONFIG_KEYSTONE_REGION=RegionOne
    CONFIG_KEYSTONE_ADMIN_TOKEN=intel
    CONFIG_KEYSTONE_ADMIN_EMAIL=root@
localhost
    CONFIG_KEYSTONE_ADMIN_USERNAME=admin
    CONFIG_KEYSTONE_ADMIN_PW=intel
    CONFIG_KEYSTONE_DEMO_PW=intel
    CONFIG_KEYSTONE_API_VERSION=v2.0
    CONFIG_KEYSTONE_TOKEN_FORMAT=UUID
    CONFIG_KEYSTONE_SERVICE_
    NAME=keystone
    CONFIG_KEYSTONE_IDENTITY_BACKEND=sql
    CONFIG_KEYSTONE_LDAP_
    URL=ldap://172.16.77.2
    CONFIG_KEYSTONE_LDAP_USER_DN=
    CONFIG_KEYSTONE_LDAP_USER_PASSWORD=
    CONFIG_KEYSTONE_LDAP_SUFFIX=
    CONFIG_KEYSTONE_LDAP_QUERY_SCOPE=one
    CONFIG_KEYSTONE_LDAP_PAGE_SIZE=-1
    CONFIG_KEYSTONE_LDAP_USER_SUBTREE=
    CONFIG_KEYSTONE_LDAP_USER_FILTER=
    CONFIG_KEYSTONE_LDAP_USER_
OBJECTCLASS=
    CONFIG_KEYSTONE_LDAP_USER_ID_
    ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_NAME_
    ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_MAIL_
    ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_ENABLED_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_ENABLED_
MASK=-1
    CONFIG_KEYSTONE_LDAP_USER_ENABLED_
DEFAULT=TRUE
```

```
    CONFIG_KEYSTONE_LDAP_USER_ENABLED_
INVERT=n
    CONFIG_KEYSTONE_LDAP_USER_ATTRIBUTE_
IGNORE=
    CONFIG_KEYSTONE_LDAP_USER_DEFAULT_
PROJECT_ID_ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_ALLOW_
CREATE=n
    CONFIG_KEYSTONE_LDAP_USER_ALLOW_
UPDATE=n
    CONFIG_KEYSTONE_LDAP_USER_ALLOW_
DELETE=n
    CONFIG_KEYSTONE_LDAP_USER_PASS_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_USER_ENABLED_
EMULATION_DN=
    CONFIG_KEYSTONE_LDAP_USER_
ADDITIONAL_ATTRIBUTE_MAPPING=
    CONFIG_KEYSTONE_LDAP_GROUP_SUBTREE=
    CONFIG_KEYSTONE_LDAP_GROUP_FILTER=
    CONFIG_KEYSTONE_LDAP_GROUP_
OBJECTCLASS=
    CONFIG_KEYSTONE_LDAP_GROUP_ID_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_GROUP_NAME_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_GROUP_MEMBER_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_GROUP_DESC_
ATTRIBUTE=
    CONFIG_KEYSTONE_LDAP_GROUP_
ATTRIBUTE_IGNORE=
    CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_
CREATE=n
    CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_
UPDATE=n
    CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_
DELETE=n
    CONFIG_KEYSTONE_LDAP_GROUP_
ADDITIONAL_ATTRIBUTE_MAPPING=
    CONFIG_KEYSTONE_LDAP_USE_TLS=n
    CONFIG_KEYSTONE_LDAP_TLS_CACERTDIR=
    CONFIG_KEYSTONE_LDAP_TLS_CACERTFILE=
    CONFIG_KEYSTONE_LDAP_TLS_REQ_
CERT=demand
    CONFIG_GLANCE_DB_PW=intel
    CONFIG_GLANCE_KS_PW=intel
    CONFIG_GLANCE_BACKEND=file
    CONFIG_CINDER_DB_PW=intel
    CONFIG_CINDER_KS_PW=intel
    CONFIG_CINDER_BACKEND=lvm
    CONFIG_CINDER_VOLUMES_CREATE=y
    CONFIG_CINDER_VOLUMES_SIZE=20G
    CONFIG_CINDER_GLUSTER_MOUNTS=
    CONFIG_CINDER_NFS_MOUNTS=
    CONFIG_CINDER_NETAPP_LOGIN=
    CONFIG_CINDER_NETAPP_PASSWORD=
    CONFIG_CINDER_NETAPP_HOSTNAME=
    CONFIG_CINDER_NETAPP_SERVER_PORT=80
    CONFIG_CINDER_NETAPP_STORAGE_
FAMILY=ontap_cluster
    CONFIG_CINDER_NETAPP_TRANSPORT_
TYPE=http
    CONFIG_CINDER_NETAPP_STORAGE_
PROTOCOL=nfs
    CONFIG_CINDER_NETAPP_SIZE_
MULTIPLIER=1.0
    CONFIG_CINDER_NETAPP_EXPIRY_THRES_
MINUTES=720
    CONFIG_CINDER_NETAPP_THRES_AVL_SIZE_
PERC_START=20
    CONFIG_CINDER_NETAPP_THRES_AVL_SIZE_
PERC_STOP=60
    CONFIG_CINDER_NETAPP_NFS_SHARES=
    CONFIG_CINDER_NETAPP_NFS_SHARES_
CONFIG=/etc/cinder/shares.conf
    CONFIG_CINDER_NETAPP_VOLUME_LIST=
    CONFIG_CINDER_NETAPP_VFILER=
    CONFIG_CINDER_NETAPP_PARTNER_
BACKEND_NAME=
    CONFIG_CINDER_NETAPP_VSERVER=
    CONFIG_CINDER_NETAPP_CONTROLLER_IPS=
    CONFIG_CINDER_NETAPP_SA_PASSWORD=
    CONFIG_CINDER_NETAPP_ESERIES_HOST_
TYPE=linux_dm_mp
    CONFIG_CINDER_NETAPP_WEBSERVICE_
PATH=/devmgr/v2
    CONFIG_CINDER_NETAPP_STORAGE_POOLS=
    CONFIG_MANILA_DB_PW=intel
    CONFIG_MANILA_KS_PW=intel
    CONFIG_MANILA_BACKEND=generic
    CONFIG_MANILA_NETAPP_DRV_HANDLES_
SHARE_SERVERS=false
    CONFIG_MANILA_NETAPP_TRANSPORT_
TYPE=https
    CONFIG_MANILA_NETAPP_LOGIN=admin
    CONFIG_MANILA_NETAPP_PASSWORD=
    CONFIG_MANILA_NETAPP_SERVER_
HOSTNAME=
    CONFIG_MANILA_NETAPP_STORAGE_
FAMILY=ontap_cluster
    CONFIG_MANILA_NETAPP_SERVER_PORT=443
    CONFIG_MANILA_NETAPP_AGGREGATE_NAME_
SEARCH_PATTERN=(.*)
    CONFIG_MANILA_NETAPP_ROOT_VOLUME_
AGGREGATE=
    CONFIG_MANILA_NETAPP_ROOT_VOLUME_
NAME=root
    CONFIG_MANILA_NETAPP_VSERVER=
    CONFIG_MANILA_GENERIC_DRV_HANDLES_
SHARE_SERVERS=true
    CONFIG_MANILA_GENERIC_VOLUME_NAME_
TEMPLATE=manila-share-%s
    CONFIG_MANILA_GENERIC_SHARE_MOUNT_
PATH=/shares

 CONFIG_MANILA_SERVICE_IMAGE_
 LOCATION=https://www.dropbox.com/s/
 vi5oeh10q1qkckh/Centos_1204_nfs_cifs.
 qcow2
    CONFIG_MANILA_SERVICE_INSTANCE_
USER=Centos
    CONFIG_MANILA_SERVICE_INSTANCE_
PASSWORD=
    CONFIG_MANILA_NETWORK_TYPE=neutron
    CONFIG_MANILA_NETWORK_STANDALONE_
```

```
GATEWAY=
    CONFIG_MANILA_NETWORK_STANDALONE_
NETMASK=
    CONFIG_MANILA_NETWORK_STANDALONE_
SEG_ID=
    CONFIG_MANILA_NETWORK_STANDALONE_
IP_RANGE=
    CONFIG_MANILA_NETWORK_STANDALONE_
IP_VERSION=4
    CONFIG_IRONIC_DB_PW=intel
    CONFIG_IRONIC_KS_PW=intel
    CONFIG_NOVA_DB_PW=intel
    CONFIG_NOVA_KS_PW=intel
    CONFIG_NOVA_SCHED_CPU_ALLOC_
RATIO=16.0
    CONFIG_NOVA_SCHED_RAM_ALLOC_
RATIO=1.5
    CONFIG_NOVA_COMPUTE_MIGRATE_
PROTOCOL=tcp
    CONFIG_NOVA_COMPUTE_MANAGER=nova.
compute.manager.ComputeManager
    CONFIG_VNC_SSL_CERT=
    CONFIG_VNC_SSL_KEY=
    CONFIG_NOVA_COMPUTE_PRIVIF=em2
    CONFIG_NOVA_NETWORK_MANAGER=nova.
network.manager.FlatDHCPManager
    CONFIG_NOVA_NETWORK_PUBIF=em1
    CONFIG_NOVA_NETWORK_PRIVIF=em3
    CONFIG_NOVA_NETWORK_
FIXEDRANGE=192.168.32.0/22
    CONFIG_NOVA_NETWORK_
FLOATRANGE=10.3.4.0/22
    CONFIG_NOVA_NETWORK_
AUTOASSIGNFLOATINGIP=n
    CONFIG_NOVA_NETWORK_VLAN_START=100
    CONFIG_NOVA_NETWORK_NUMBER=1
    CONFIG_NOVA_NETWORK_SIZE=255
    CONFIG_NEUTRON_KS_PW=intel
    CONFIG_NEUTRON_DB_PW=intel
    CONFIG_NEUTRON_L3_EXT_BRIDGE=br-ex
    CONFIG_NEUTRON_METADATA_PW=intel
    CONFIG_LBAAS_INSTALL=n
    CONFIG_NEUTRON_METERING_AGENT_
INSTALL=n
    CONFIG_NEUTRON_FWAAS=n
    CONFIG_NEUTRON_ML2_TYPE_
DRIVERS=vxlan
    CONFIG_NEUTRON_ML2_TENANT_NETWORK_
TYPES=vxlan
    CONFIG_NEUTRON_ML2_MECHANISM_
DRIVERS=openvswitch
    CONFIG_NEUTRON_ML2_FLAT_NETWORKS=*
    CONFIG_NEUTRON_ML2_VLAN_RANGES=physn
et1,physnet2,physnet3
    CONFIG_NEUTRON_ML2_TUNNEL_ID_RANGES=
    CONFIG_NEUTRON_ML2_VXLAN_
GROUP=239.1.1.100
    CONFIG_NEUTRON_ML2_VNI_
RANGES=1001:2000
    CONFIG_NEUTRON_L2_AGENT=openvswitch
    CONFIG_NEUTRON_LB_INTERFACE_
MAPPINGS=
    CONFIG_NEUTRON_OVS_BRIDGE_
MAPPINGS=physnet1:br-ex,physnet2:br-
mng,physnet3:br-em3
    CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-
ex:em1,br-mng:em2,br-em3:em3
    CONFIG_NEUTRON_OVS_TUNNEL_IF=
    CONFIG_NEUTRON_OVS_VXLAN_UDP_
PORT=4789
    CONFIG_HORIZON_SSL=n
    CONFIG_HORIZON_SECRET_KEY=dd5a2abbce
f747f7a7bafede42947d71
    CONFIG_HORIZON_SSL_CERT=
    CONFIG_HORIZON_SSL_KEY=
    CONFIG_HORIZON_SSL_CACERT=
    CONFIG_SWIFT_KS_PW=intel
    CONFIG_SWIFT_STORAGES=
    CONFIG_SWIFT_STORAGE_ZONES=1
    CONFIG_SWIFT_STORAGE_REPLICAS=1
    CONFIG_SWIFT_STORAGE_FSTYPE=ext4
    CONFIG_SWIFT_HASH=c2a8ece9563b4666
    CONFIG_SWIFT_STORAGE_SIZE=2G
    CONFIG_HEAT_DB_PW=intel
    CONFIG_HEAT_AUTH_ENC_
KEY=eb12297f095c4958
    CONFIG_HEAT_KS_PW=intel
    CONFIG_HEAT_CLOUDWATCH_INSTALL=n
    CONFIG_HEAT_CFN_INSTALL=n
    CONFIG_HEAT_DOMAIN=heat
    CONFIG_HEAT_DOMAIN_ADMIN=heat_admin
    CONFIG_HEAT_DOMAIN_PASSWORD=intel
    CONFIG_PROVISION_DEMO=n
    CONFIG_PROVISION_TEMPEST=n
    CONFIG_PROVISION_DEMO_
FLOATRANGE=172.24.4.224/28
    CONFIG_PROVISION_IMAGE_NAME=cirros
    CONFIG_PROVISION_IMAGE_URL=http://
download.cirros-cloud.net/0.3.3/cirros-
0.3.3-x86_64-disk.img
    CONFIG_PROVISION_IMAGE_FORMAT=qcow2
    CONFIG_PROVISION_IMAGE_SSH_
USER=cirros
    CONFIG_PROVISION_TEMPEST_USER=
    CONFIG_PROVISION_TEMPEST_USER_
PW=intel
    CONFIG_PROVISION_TEMPEST_
FLOATRANGE=172.24.4.224/28
    CONFIG_PROVISION_TEMPEST_REPO_
URI=https://github.com/openstack/tempest.
git
    CONFIG_PROVISION_TEMPEST_REPO_
REVISION=master
    CONFIG_PROVISION_ALL_IN_ONE_OVS_
BRIDGE=n
    CONFIG_CEILOMETER_
SECRET=54188c6a86154776
    CONFIG_CEILOMETER_KS_PW=intel
    CONFIG_CEILOMETER_COORDINATION_
BACKEND=redis
    CONFIG_MONGODB_HOST=172.16.77.2
    CONFIG_REDIS_MASTER_HOST=172.16.77.2
    CONFIG_REDIS_PORT=6379
    CONFIG_REDIS_HA=n
    CONFIG_REDIS_SLAVE_HOSTS=
```

```
    CONFIG_REDIS_SENTINEL_HOSTS=
    CONFIG_REDIS_SENTINEL_CONTACT_HOST=
    CONFIG_REDIS_SENTINEL_PORT=26379
    CONFIG_REDIS_SENTINEL_QUORUM=2
    CONFIG_REDIS_MASTER_NAME=mymaster
    CONFIG_SAHARA_DB_PW=intel
    CONFIG_SAHARA_KS_PW=intel
    CONFIG_TROVE_DB_PW=intel
    CONFIG_TROVE_KS_PW=intel
    CONFIG_TROVE_NOVA_USER=trove
    CONFIG_TROVE_NOVA_TENANT=services
    CONFIG_TROVE_NOVA_PW=intel
    CONFIG_NAGIOS_PW=intel
```

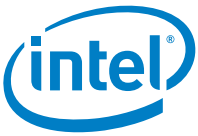
Appendix B: Abbreviations

ABBREVIATION	DESCRIPTION
CDN	Content Delivery Network
CGNAT	Carrier-Grade Network Address Translation
CPU	Central Processing Unit
DDoS	Distributed DoS
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
GGSN	Gateway GPRS Support Node
Gi-LAN	Gateway-Internet LAN
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HD	Hard Disk
HTTP	Hypertext Transfer Protocol

ABBREVIATION	DESCRIPTION
IDE	Integrated Drive Electronics
LAN	Local Area Network
NFV	Network Functions Virtualization
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
PTS	Policy Traffic Switch
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
SDE	Service Delivery Engine
SPB	Subscriber Policy Broker
TDF	Traffic Detection Function
VLAN	Virtual LAN
VM	Virtual Machine
VNF	Virtualized Network Functions

Appendix C: References

REFERENCE	SOURCE
Brocade 5600 vRouter Data sheet	http://www.brocade.com/content/dam/common/documents/content-types/datasheet/brocade-vrouter-ds.pdf
Evaluating Dynamic Service Function Chaining for the Gi-LAN White Paper	http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/evaluating-dynamic-service-function-chaining-for-the-gilan-paper.pdf
F5 BIG-IP Carrier-Grade NAT Data Sheet	http://www.f5.com/pdf/products/big-ip-cgnat-datasheet.pdf
F5 BIG-IP Advanced Firewall Manager Data Sheet	http://www.f5.com/pdf/products/big-ip-advanced-firewall-manager-datasheet.pdf
Open vSwitch	http://openvswitch.org/
Sandvine Policy Traffic Switch Virtual Series	https://www.sandvine.com/platform/policy-traffic-switch/pts-virtual-series.html
Sandvine Service Delivery Engine Virtual Series	https://www.sandvine.com/platform/service-delivery-engine.html
Sandvine Subscriber Policy Broker Virtual Series	https://www.sandvine.com/downloads/general/platform/subscriber-policy-broker/sandvine-subscriber-policy-broker.pdf
Snort	https://www.snort.org/ https://www.snort.org/downloads/archive/snort/snort-2.9.6.1.tar.gz
Apache HTTP Server	https://httpd.apache.org/ https://archive.apache.org/dist/httpd/
Apache Traffic Server	http://trafficserver.apache.org/ https://www.snort.org/downloads/archive/snort/snort-2.9.6.1.tar.gz



 1116/MH/MESH/PDF 335266-001US