Using Tacker + ODL-SFC for NFV Management and Orchestration with NSH

**Document History**

|  |  |  |
| --- | --- | --- |
| Victor Mehmeri | 6/30/2016 | Initial Version |

**Summary**

Tacker VM Setup

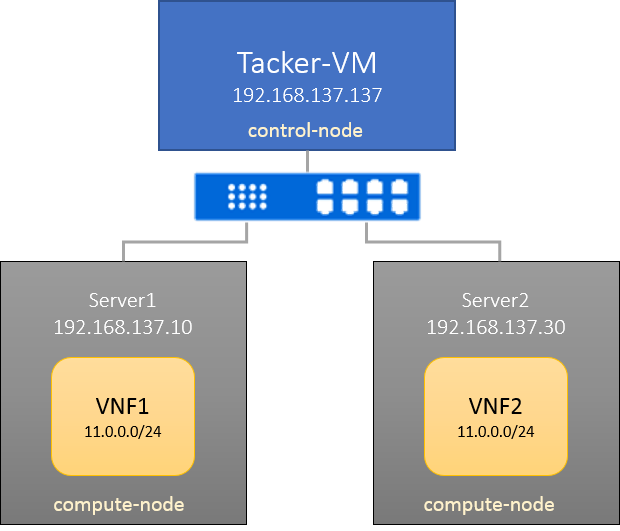
Compute Node Setup

Demo Walkthrough

Troubleshooting and Cleanup

Hacking

**Architecture**

****

# Tacker VM Setup

Do a fresh install of Ubuntu 14.04 on a VirtualBox VM with at least 4GB memory, 20G disk, and then follow the steps below:

Create stack user (if you haven’t set up the virtual machine with stack user already):

sudo useradd -m stack

sudo passwd stack stack

*(password will be stack)*

Give the user passwordless sudo*:*

sudo visudo –f /etc/sudoers.d/override

*add the following line:*

stack ALL=(ALL) NOPASSWD: ALL

*ctrl+o to save, then ctrl+x to exit*

Login as stack user *(‘su stack’*), and run the following to install OVS with NSH support*:*

sudo apt-get update –y

sudo apt-get install autoconf libtool git curl –y

*curl* [*https://raw.githubusercontent.com/priteshk/ovs/nsh-v8/third-party/start-ovs-deb.sh*](https://raw.githubusercontent.com/priteshk/ovs/nsh-v8/third-party/start-ovs-deb.sh)*| bash*

*After installation is complete, run:*

git clone -b stable/liberty <https://github.com/trozet/devstack>

git clone <https://github.com/trozet/sfc-random.git>

git clone <https://github.com/vmehmeri/devstack-lab.git>

cp devstack-lab/control.conf devstack/local.conf

cd devstack

sed -i 's/192.168.137.137/<your IP>/g' local.conf

*The last command is to update the conf file with your IP address*

*Finally, run:*

./stack.sh

*This will take a while, as the this script is going to download all dependencies for Openstack and Tacker. After it finishes, you can uncomment the following line:*

#OFFLINE=True

*So next time you run it, it won’t re-download everything and will use what you already have locally. Comment this line again if you need to update things.*

# Compute Node Setup

The following should be done for each compute node.

Run the same steps above from a fresh Ubuntu 14.04 install to create stack user, give it passwordless sudo access, and install OVS-NSH. From then:

*After installation is complete, run:*

git clone -b stable/liberty <https://github.com/trozet/devstack>

git clone <https://github.com/vmehmeri/devstack-lab.git>

cp devstack-lab/compute.conf devstack/local.conf

cd devstack

sed -i 's/1.2.3.4/<CONTROLLER-IP>/g' local.conf

sed -i 's/0.0.0.0/<YOUR-IP>/g' local.conf

*The last command is to update the conf file with the compute node’s IP address, and is the only step that will differ in each compute node setup.*

*Inspect local.conf file to see if you have the correct IP addresses: SERVICE\_HOST is the control node, HOST is the compute node you’re configuring:*

HOST\_IP=192.168.137.10

MULTI\_HOST=1

SERVICE\_HOST=192.168.137.137

***Wait until the control node stack script is complete****, and then run:*

./stack.sh

*After it finishes, you can also uncomment the OFFLINE=True line. Re-running it with the OFFLINE mode will be quite fast. Finally, run:*

cp ../devstack-lab/vxlan\_workaround.sh .

./vxlan\_workaround.sh

*(this doc describes what this workaround is for:* [*https://docs.google.com/document/d/1NMq5oudRcXCm-n9ALjdEVPIvUDZhgzq\_OFu6Pvf9E8E/edit#heading=h.2qj6pt3dgetw*](https://docs.google.com/document/d/1NMq5oudRcXCm-n9ALjdEVPIvUDZhgzq_OFu6Pvf9E8E/edit#heading=h.2qj6pt3dgetw)*)*

# Demo Walkthrough

After all the compute nodes are up and running, open a terminal in the **control node** and follow the steps below:

cd devstack

cp accrc/admin/admin accrc/admin/heat

source accrc/admin/heat

wget <https://www.dropbox.com/s/focu44sh52li7fz/sfc_cloud.qcow2>

neutron net-create net\_mgmt --provider:network\_type=vxlan --provider:segmentation\_id 1005

neutron subnet-create net\_mgmt 11.0.0.0/24

openstack image create sfc --public --file ./sfc\_cloud.qcow2

openstack flavor create custom --ram 1000 --disk 5 –public

tacker vnfd-create --vnfd-file ../sfc-random/test-vnfd.yaml

tacker vnf-create --name testVNF1 --vnfd-name test-vnfd

This will instantiate the VNF. Note it may take several minutes to boot. You can check after a few minutes if the VNF is ACTIVE:

tacker vnf-list

If you have ERROR, try deleting the vnf and re-creating it:

tacker vnf-delete testVNF1

tacker vnf-create --name testVNF1 --vnfd-name test-vnfd

Then create the second one:

tacker vnf-create --name testVNF2 --vnfd-name test-vnfd

If the compute nodes have the same resources (CPU, RAM), then the second VNF will be spawn the other node, since it naturally have more resources available (Openstack-nova primarily takes into consideration available RAM). If that is not the case and you want to force placement in the second compute node, one way to do it is by creating a new template:

cp ../sfc-random/test-vnfd.yaml test-vnfd2.yaml

edit the file and change the following line:

availability zone: nova

to:

availability zone: nova2

Login to Horizon (Web UI) by typing the controller’s IP address into a web browser, username “heat” and password “devstack”

Navigate to System -> Host Aggregates

Click on:



Give it a name, and below “Availability Zone” type “nova2”

Select “Manage Hosts within Aggregate” tab, and click on the ‘+’ sign next to the second compute node’s hostname.

Now run:

tacker vnf-create --name testVNF2 --vnfd-name test-vnfd2

And VNF will be created in the second host.

Finally, to setup the Service Function Chain, run:

tacker sfc-create --name mychain --chain testVNF1,testVNF2

tacker sfc-classifier-create --name myclass --chain mychain --match dest\_port=80,protocol=6

In the compute nodes, check OVS to see if you have SFF and Classifier flows:

sudo ovs-ofctl dump-flows br-int –OOpenFlow13 | grep tp\_dst=80

sudo ovs-ofctl dump-flows br-int –OOpenFlow13 | grep nsp=

(if no lines are printed, something went wrong).

**Testing SFC**

1. Create Client and Server instances

Using either Horizon or CLI create a cirros instance named http\_client, with default flavor (m1.tiny). For creating HTTP server you can create another instance using the sfc image (custom flavor). Use following usernames and passwords:

* 1. [**cirros**] username: cirross ; password: cubswin:)
  2. [**sfc**] username: root ; password: octopus

1. Open http\_server console via Horizon and run:

service iptables stop

python -m SimpleHTTPServer 80

1. Open tesVNF1 console via Horizon and run (same for testVNF2):

python ~/vxlan\_tool/vxlan\_tool.py -i eth0 -d forward -v on

1. Finally, login to http\_client and run:

curl <http server ip>

Verify you see packets being redirected through the SFC (hitting the VNF in the horizon console)

Verify you receive 200 OK response from the http server

# Troubleshooting and Cleanup

Tacker logs:

/opt/stack/logs/tacker.log

ODL logs:

/opt/stack/logs/screen-karaf.txt

Attaching to screens:

screen -r

CTRL+a+n : next screen

CTRL+a+p : previous screen

CTRL+c : stop service

To restart a service: CTRL + C, Up Arrow (will show last command) + Enter

CTRL+a+d: detach

Unstack:

./unstack.sh

(Sometimes it is necessary to reboot the machine before running stack.sh again in the control node).

Clean up (remove all databases, uninstall packages, etc):

./clean.sh

# Hacking

Useful codes to look into:

# Tacker plugin

/opt/stack/tacker/tacker/plugin.py

# Tacker ODL SFC driver

/opt/stack/tacker/tacker/sfc/drivers/opendaylight.py

# Networking ODL plugin directory

/opt/stack/networking-odl

# References

[1] <https://github.com/trozet/sfc-random/blob/master/tacker_sfc_walkthrough.txt>

[2] [https://wiki.**openstack**.org/wiki/**Tacker**](https://wiki.openstack.org/wiki/Tacker)

[3] <https://www.ietf.org/proceedings/93/slides/slides-93-nfvrg-25.pdf>

[4] <http://tacker-docs.readthedocs.io/en/latest/>