Felhő alapú hálózatok

OpenStack Neutron Networking gyakorlat
IB.213

Indítás: Debian Jessie DevStack experimental...
Live rendszer, azaz újraindítás után minden elveszik. Célszerű a gyakorlat jegyzőkönyvét online dokumentumba szerkeszteni, csoportonként egyet!
3 fős csoportok: 1 controller, 1 network, 1 compute

stack felhasználóra váltás:
`su - stack`
jelszó: stack

A stack felhasználónak a sudo engedélyezett, ha valamit root felhasználóként lehet csak végrehajtani.

Mindenek előtt adjuk ki a következő utasítást (a Debian a DevSatck és az SSL3 nem akart együttműködni…)
sed -i '/"sslv3": ssl.PROTOCOL_SSLv3/c\#"sslv3": ssl.PROTOCOL_SSLv3' /opt/stack/nova/nova/openstack/common/sslutils.py

1. feladat: 3 node-os DevStack rendszer elindítása

A gépeken egy interfész a labor hálózatra van kötve, és 192.168.213.X címet kap, ezen keresztül van internet elérés, ehhez ne nyúljunk, ez általában az eth1, de ezt ellenőrizzük!!!
A menedzsment hálózat a 192.168.42.0/24 lesz egy másik interfészen, ami általában az eth0, de ezt is ellenőrizd! (Ez a szürke kábel, ami ezúttal előre össze lesz kötve a patch panelen egy switch-en keresztül, de a kapcsolat megléte itt is ellenőrizendő!)

A VXLAN tunnelekhez nem lesz külön hálózati interfész, hanem újrahasznosítjuk a menedzsment hálózatot, afelett lesznek kihúzva.

Hálózati beállítások és local.conf fájlok az egyes node-okhoz:
local.conf helye: /home/stack/devstack könyvtárban

CONTROLLER:
Menedzsment hálózati interfész beállítása:
sudo ifconfig eth0 192.168.42.11/24

local.conf:
[[local|localrc]]
HOST_IP=192.168.42.11
ADMIN_PASSWORD=secret
MYSQL_PASSWORD=secret
RABBIT_PASSWORD=secret
SERVICE_PASSWORD=secret
SERVICE_TOKEN=tokentoken
MULTI_HOST=1
LOGFILE=/opt/stack/logs/stack.sh.log
disable_service n-cpu
disable_service n-net
enable_service q-svc
enable_service neutron
ENABLE_TENANT_TUNNELS=True
Q_PLUGIN=ml2
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_ML2_PLUGIN_VXLAN_TYPE_OPTIONS=(vni_ranges=400:500)

NETWORK:
Menedzsment hálózati interfész beállítása:
sudo ifconfig eth0 192.168.42.21/24

local.conf:
[[local|localrc]]
HOST_IP=192.168.42.21 # replace this with the IP address of the network node
SERVICE_HOST=192.168.42.11
MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
Q_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
LOGFILE=/opt/stack/data/stack.log
SCREEN_LOGDIR=/opt/stack/data/log
ENABLED_SERVICES=q-agt,q-l3,q-dhcp,q-meta,rabbit
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_ML2_PLUGIN_VXLAN_TYPE_OPTIONS=(vni_ranges=400:500)
ENABLE_TENANT_TUNNELS=True
MYSQL_PASSWORD=secret
RABBIT_PASSWORD=secret
ADMIN_PASSWORD=secret
SERVICE_PASSWORD=secret
SERVICE_TOKEN=tokentoken

COMPUTE:
Menedzsment hálózati interfész beállítása:
ifconfig eth0 192.168.42.31/24

local.conf:

```
[[local|localrc]]
HOST_IP=192.168.42.31 # replace this with the IP address of the compute node
ENABLED_SERVICES=n-cpu,rabbit,neutron,q-agt
LOGFILE=/opt/stack/data/stack.log
SCREEN_LOGDIR=/opt/stack/data/log
# Openstack services running on controller node
SERVICE_HOST=192.168.42.11 # replace this with the IP address of the controller node
MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
MYSQL_PASSWORD=secret
RABBIT_PASSWORD=secret
Q_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
ENABLE_TENANT_TUNNELS=True
Q_PLUGIN=ml2
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_ML2_PLUGIN_VXLAN_TYPE_OPTIONS=(vni_ranges=400:500)
Q_USE_SECGROUP=True
KEYSTONE_AUTH_HOST=$SERVICE_HOST
ADMIN_PASSWORD=secret
SERVICE_PASSWORD=secret
SERVICE_TOKEN=tokentoken
NOVA_VNC_ENABLED=True
NOVNCPROXY_URL="http://$SERVICE_HOST:6080/vnc_auto.html"
VNCSERVER_LISTEN=$HOST_IP
VNCSERVER_PROXYCLIENT_ADDRESS=$VNCSERVER_LISTEN
```

Ezután a ./stack.sh -val elindítani minden node-ot. (Pár perc is lehet, főleg a controlleren.)

2. feladat:
Hozzuk létre a virtuális hálózatokat az általános leírás alapján, figyelembe véve a labor környezetet! Lehet az alábbi parancssoros módszerrel, vagy a dashboard felülelen is kattingatva.
http://docs.openstack.org/juno/install-guide/install/apt/content/neutron-initial-networks.html
(External és Tenant -ra kattintva találhatóak a leírások)
source admin-openrc.sh helyett: `source openrc admin admin`
illetve `source openrc demo demo`
ha admin, illetve demo user-ként akarunk tevékenykedni parancssorból!

Labor net: 192.168.213.0/24
Gateway: 192.168.213.1
Floating IP tartományok, ezek most nem keveredhetnek a csoportok között, mert a külső laborhálózatból allokáljuk:
/29-es netmask

Csoport	Network#	IP range	Bcast
Fal melletti 1-3 gép csoport	.200	.201-.206	.207
Fal melletti 4-6 gép csoport	.208	.209-.214	.215
Fal melletti 7-9 gép csoport	.216	.217-.222	.223
Fal melletti 10-12 gép csoport	.224	.225-.230	.231
Középső 2-4 gép csoport	.232	.233-.238	.239

A horizon dashboard “Network topology”-ban tekintsük meg mi jött létre!

Ahhoz, hogy elérjük a labor hálózatot a VM-ekből a Neutron network node (általában eth2, de ezt ellenőrizd!) IP cím nélküli (unnumbered) interfészét be kell kötni a patch panelen a labor hálózat kapcsolójába (amibe a kék kábelek mennek), és hozzáadni a külső bridge-hez a network node-on kiadva az alábbi parancsot:
`#sudo ovs-vsctl add-port br-ex` *`INTERFACE_NAME`*

Indítsunk el 2 CirrOS VM-et a horizon dashboard felületről, úgy hogy a korábban létrehozott demo-net -hez kapcsolódjanak!
Ellenőrizzük a következőket és a parancsok eredményét, kimenetét mentsétek el a jegyzőkönyvbe: (A horizon dashboardon az Instance Console felületen tudunk belépni a VM-ekre. user: “cirros”, pw: “cubswin:)”)
2.1 a két VM látja egymást a hálózaton (ping)
2.2 a VM-ből elérhető a külső hálózat (ping külső IP cím)

3. Az további kérdésekre adott válaszokat, a parancsok eredményét, kimenetét mentsétek el a jegyzőkönyvbe!

Ezekkel az utasításokkal lehet az OpenvSwitch (ovs) -re vonatkozó információkat lekérdezni (sudo is kell), további segítség: man ovs-vsctl vagy ovs-vsctl --help:

```
# ovs-vsctl show
# ovs-vsctl list bridge
# ovs-vsctl list bridge <ovs-bridge name>
#ovs-vsctl list port
# ovs-vsctl list port <ovs-port name>
```

Az áttekintéshez segít a csütörtöki fóliasor 12. oldala.
3.1 Milyen virtuális kapcsolók vannak a **network** és **compute** node-on?
3.2 Milyen portok vannak az egyes virtuális kapcsolókon a **compute** node-on?
3.3 A **compute** node-on: a br-int -en és a br-tun-on mely portok standard Linux portok, amelyek közvetlenül elérhetők (pl. ifconfig paranccsal) és melyek OpenvSwitch belső portok, amelyek nem?
3.4 Milyen iptables szabályok vonatkoznak a tap device-ra a **compute** node-on? (iptables -S)
3.5 Milyen br-tun transzformációs szabályok vannak VM indítás után a compute node-on? Az OpenFlow protokollon keresztül beállított szabályok lekérdezése:

```
#ovs-ofctl dump-flows br-tun
```

Melyek ezek közül, amelyek a tunnelingre vonatkoznak?

4. Hálózati névterek (Csütörtöki fóliasor 8. oldal.)
4.1 Milyen névterek jöttek létre a **network** node-on? (ip netns)
4.2 Mi az interfész konfiguráció a DHCP szerver névtérben a **network** node-on?

```
# ip netns exec ...
```

4.3 Mi az interfész konfiguráció a router névtérben a **network** node-on?

5. A csomagok útja

5.1 Milyen portokon halad keresztül egy VM-VM közötti forgalom?

5.2 Milyen portokon halad keresztül egy VM-külső hálózat közötti forgalom?

5.3 Azonosítsuk be tcpdump-pal a VXLAN tunnel forgalmat azon a hálózati interfészen, ahová fel van húzva (általában eth0)! Mentsük el fájlba a teljes csomagokat (-s 0) és nézzük meg a VXLAN csomagok felépítését wiresharkkal is! (VXLAN-ról 7. előadás fóliáiban)

6. Rendeljünk egy VM-hez Floating IP-t a horizon dashboradon keresztül! Ahhoz, hogy át legyen engedve a forgalom, a tűzfal szabályokat be kell állítani az alábibak szerint:
http://docs.openstack.org/user-guide/content/Launching_Instances_using_Dashboard.html
“Add a rule to the default security group” alatti pontok
Teszteljük hogy a VM valamelyik node-ról elérhető-e (pl. ping vagy ssh) a Floating IP címen!