Potek naloge 2

V okviru te naloge bomo namestili vse komponente OpenStack na en fizični strežnik (*ang. one node installation*), kar je za naše potrebe in razumevanje nastavitev platforme povsem dovolj. Postopek je povzet po dokumentaciji namestitev in nastavitev platforme OpenStack. Tekom naloge bomo namestili komponente, kot je prikazano na Sliki 1.

Slika 1: Namestitev platforme na en fizični strežnik.

Najprej posodobimo seznam nameščenih programskih paketov in nato namestimo njihove posodobitve. Tako si zagotovimo pravilno delujočo programsko opremo:

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
```

Med nadgrajevanjem programske opreme nas vpraša, če želimo namestiti novi zaganjalnik Grub. Pri tem izberemo možnost, da obdržimo trenutnega.

V kolikor želimo, da se vse spremembe uveljavijo, bomo ponovno zagnali OS z ukazom:

$sudo reboot

Ponovno vnesem prijavne podatke za okolje Ubuntu. Po ponovnem zagonu bi se morala izvajati MySQL in RabbitMQ, tako da ni potrebno zaganjati teh procesov.

Komponente bomo namestili in nastavili v naslednjem zaporedju: Keystone, Glance, Nova in Cinder.

Pri nameščanju prve komponente je podan natančen opis kako bomo kaj namestili; enak opis velja tekom nameščanja ostalih komponent, vendar ni natančno podan. V kolikor se pri namestitvi in nastavitvi določene komponente pojavijo nastavitve dodatnih parametrov, jih bomo natančno opisali.

Namestitev in nastavitev komponente za identifikacije ter avtorizacijske storitve

Najprej namestimo Keystone, storitev za avtentikacijo in identifikacijo, z ukazom:

$ sudo apt-get install keystone

Po vsaki namestitvi posamezne komponente je priporočljivo preveriti, če se je pravilno namestila. V primeru neuspešne namestitve komponente nas tolmač ukazov obvesti o napaki oziroma v določenih primerih tudi sporoči, kaj je narobe.

V datoteki */etc/keystone/keystone.conf* pod razdelkom *[database]* nastavimo parameter *connection*, v katerem se nahajajo prijavni podatki storitve in kje se nahaja podatkovna baza MySQL. V spodnjem primeru je uporabnik *keystoneUser*, geslo je *keystoneDBPass*, naslov, kjer se nahaja podatkovna baza, je *<ime_našega_strežnika>* in ime podatkovne baze *keystoneDatabase*:

connection=mysql://keystoneUser:keystoneDBPass@<ime_nasega_strežnika>/keystoneDatabase

 Zapomniti si je potrebno prijavne podatke storitve Keystone za dostop do podatkovne baze.

Med namestitvijo komponente se tvori privzeta podatkovna baza SQLite, ki jo bomo izbrisali z ukazom in ponovno zagnali proces komponente Keystone:

$ sudo rm /var/lib/keystone/keystone.db

$ sudo service keystone restart

Če smo komponento pravilno namestili, preverimo z ukazom:

$ sudo service keystone status

V kolikor na seznamu ni imena komponente ali procesov komponente, potem smo naredili napako pri nastavitvah in je nastavitev potrebno popraviti. Procese komponente bomo prepoznali kot *<ime_komponente>-<API_komponente>*.

 To velja za vse komponente OpenStacka v tem tečaju.

Keystone uporablja podatkovno bazo MySQL za uporabnike, vloge, stanovalce, procese in omejitve stanovalcev. Zato je potrebno ustvariti podatkovno bazo in storitvi dodeliti pravice dostopa. Najprej se je potrebno prijaviti v konzolo MySQL-a kot korenski uporabnik:

$ mysql -u root –p *//se prijavimo v konzolo MySQL*

//in vnesemo geslo, ki smo ga uporabili pri prvi nalogi

mysql>CREATE DATABASE keystoneDatabase; // nato ustvarimo podatkovno bazo

//keystoneDatabase

mysql>GRANT ALL PRIVILEGES ON keystoneDatabase. TO 'keystoneUser'@'<ime_našega_strežnika>' IDENTIFIED BY 'keystoneDBPass';*

//nastavimo najvišje pravice dostopa uporabnika keystoneUser z geslom //keystoneDBPass do podatkovne baze keystoneDatabase

mysql>GRANT ALL PRIVILEGES ON keystoneDatabase. TO 'keystoneUser'@'%' IDENTIFIED BY 'keystoneDBPass';*

//podobno kot v prejšnjem ukazu, vendar smo pri tem prepričani, da bo pri tem sprejel //posebne znake, ki se lahko pojavijo v imenu računalnika

mysql>exit //zapustimo konzolo

Po ustvarjanju prijave komponente v podatkovni bazi sinhroniziramo komponento in podatkovno bazo z ukazom:

$ sudo keystone-manage db_sync

Tako preverimo, če smo komponenti pravilno nastavili dostop do podatkovne baze. Pri tem se samodejno tvorijo vse potrebne tabele, ki jih Keystone pri svojih opravilih uporablja.

 V kolikor nam tolmač ukazov vrne napako, potem smo v podatkovni bazi vpisali nepravilne podatke.

Namestitev in nastavitev komponente za shranjevanje diskovnih slik

Ponovno kot prej bomo namestili komponento, spremenili parametre v nastavitveni datoteki in preverili delovanje komponente. S spodnjim ukazom bomo namestili komponento Glance:

$ sudo apt-get install glance python-glanceclient

Komponenta v podatkovni bazi MySQL shranjuje podatke o slikah diskov. Celoten postopek je precej podoben kot pri namestitvi prejšnje komponente. Najprej nastavimo povezavo na bazo, nastavimo potrebne parametre v nastavitvenih datotekah in preverimo delovanje. V `/etc/glance/glance-api.conf`in `/etc/glance/glance-registry.conf` pod razdelkom *[databse]* nastavimo povezavo:

connection=mysql://glanceUser:glanceDBPass@<_ime_nasega_racunalnika>/glanceDatabase

 Zapomniti si je potrebno prijavne podatke storitve Glane za dostop do podatkovne baze.

Kot smo že prej omenili, je potrebno, da se mora vsaka storitev tekom svojih opravil najprej avtenticirati. To nastavimo v datotekah `/etc/glance/glance-api.conf` in `/etc/glance/glance-registry.conf`, pod razdelkom *[keystone_authtoken]* in

[paste_deploy]. Avtetikacija se izvaja na vratih 5000, gostitelj je *<ime_našega_strežnika>*, na portu 35357 se izvaja avtorizacija:

[keystone_authtoken] //razdelek v nastavitveni datoteki
auth_uri = http://<ime_nasega_strežnika>:5000 //dopišemo auth_uri, ker ga prej
//ni bilo v tem razdelke, na vratih 5000 se izvaja avtorizacija preko naslova URI
auth_host = <ime_našega_strežnika> //ime gostitelje avtorizacije v tem
//primeru ime našega strežnika
auth_port = 35357 //na vratih 35357 se izvaja avtentikacija
auth_protocol = http //avtentikacija se izvaja preko protokola HTTP
admin_tenant_name = service //ime stanovalca je storitevhtt
admin_user = glanceService //ime procesa nastavimo na
//glanceService
admin_password = glancePass //geslo procesa

[paste_deploy] //razdelek v nastavitveni datoteki
flavor=keystone //storitev identifiakcije skrib za predloge navideznih
//strojev

 Zapomniti si je potrebno prijave podatke soritve Glance.

 Trenutno nastavljeni prijavni podatki se bodo kasneje uporabili pri kreiranju procesa in stanovalca.

Po opravljeni nastavitvi ponovno zaženemo procesa:

$ sudo service glance-registry restart
$ sudo service glance-api restart

 V kolikor se kateri izmed procesov ne zažene, smo naredili napako v konfiguracijah.

Ustvarimo bazo, ki jo bo komponenta uporabljala:

$ mysql -u root -p

mysql> CREATE DATABASE glanceDatabase;

mysql> GRANT ALL PRIVILEGES ON glanceDatabase. TO 'glanceUser'@<ime_našega_Strežnik> IDENTIFIED BY 'glanceDBPasspassdb';*

mysql> GRANT ALL PRIVILEGES ON glanceDatabase. TO 'glanceUser'@'%' *

IDENTIFIED BY glanceDBPass';

mysql> exit

Opravimo sinhronizacijo s komponento, pri kateri se vzpostavi tvorjenje tabel podatkovne baze *glanceDatabse*:

$ sudo glance-manage db_sync

Odstranimo še privzeto podatkovno bazo SQLite:

$ sudo rm /var/lib/glance/glance.sqlite

 V nekaterih primerih baza SQLite ni privzeta; potem je ni potrebno odstranjevati.

Namestitev in nastavitve računske komponente

Postopek je podoben kot prej. Najprej je potrebno namestiti vse potrebne procese Nove, ji omogočiti dostop do baze in jo ustrezno nastaviti:

$ sudo apt-get install nova-api nova-cert nova-conductor nova-consoleauth nova-novncproxy nova-scheduler python-novaclient nova-compute-kvm nova-network

V nastavitveni datoteki */etc/nova/nova.conf* ustvarimo razdelek *[database]* in nastavimo povezavo na bazo:

[database]

connection = mysql://novaUser:novaDBPass@<ime_našega_računalnika>/novaDatabase

 Zabeležiti si je potrebno prijavne podatke, ki jih Nova uporablja za dostop do baze.

V enaki nastavitveni datoteki je računskim storitvam potrebno nastaviti tudi dostop do sporočilnega strežnika RabbitMQ. Pod razdelek *[DEFAULT]* je potrebno namestiti, kateri sporočilni strežnik se uporablja, kje se le-ta nahaja in geslo za dostop. Dodali bomo nastavitve:

rpc_backend = rabbit *//za sporočilni strežnik nastavimo rabbit*

rabbit_host = <ime_našega_strežnika> *//nastavimo gostitelja srežnika*

rabbit_password = rabbitpass *//za dostop storitve do sporočilnega //strežnika moramo nastaviti geslo*

 Zabeležiti si je potrebno geslo za dostop do sporočilnega strežnika.

Prav tako je v enaki nastavitveni datoteki potrebno nastaviti IP naslove, na katerih se lahko izvajajo storitev oddaljenega dostopa do komponente. Dodali bomo naslednje nastavitve:

[DEFAULT]

my_ip = 192.168.200.26 *//nastavimo naslov IP našega računalnika*

vncserver_listen = 192.168.200.26 *//nastavim IP naslov, kjer lahko poteka oddaljeni dostop*

vncserver_proxyclient_address = 192.168.200.26 *//nastavimo IP naslov proxy //strežnika*

vnc_enabled =True *//omogočimo oddaljeni dostop*

novncproxy_base_url = http://<ime_našega_strežnika>:6080/vnc_auto.html

glance_host = <ime_našega_strežnika> //nastavimo ime gostitelja komponente //Glance

 Celoten izsek je potrebno dodati pod razdelek *[DEFAULT]*.

V nastavitveno datoteko pod razdelek *[DEFAULT]* je potrebno nastaviti, da se bo storitev preko storitve Keystona avtenticirala:

[DEFAULT]
auth_strategy = keystone //storitev se avtenticira preko storitve Keystone

Nato dodamo razdelek *[keystone_authtoken]*, kamor vnesemo prijavne podatke storitve. Opis parametrov je enak prejšnjemu.

[keystone_authtoken]
auth_uri = http://<ime_našega_Strežnika>:5000
auth_host = <ime_našega_strežnika>
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = novaService
admin_password = novaPass

 Potrebno je odstraniti privzeto podatkovno bazo *nova.sqlite*. V nekaterih primerih se privzeta baza ne ustvari. Odstranimo jo s ukazom:

$ sudo rm /var/lib/nova/nova.sqlite

Potem še preverimo pravilnost nastavljenih parametrov v nastavitvenih datotekah. Če se vsi procesi ponovno zaženejo, potem smo jih pravilno nastavili:

$sudo service nova-api restart

$sudo service nova-cert restart
$sudo service nova-consoleauth restart
$sudo service nova-scheduler restart
$sudo service nova-conductor restart
$sudo service nova-novncproxy restart
$sudo service nova-compute restart
$sudo service nova-network restart

 V kolikor se določeni proces ne zažene ponovno, potem je napaka v konfiguracijah.

Ponovno tvorimo podatkovno bazo, ki jo bo uporabljala storitev Nova. Ime podatkovne baze nastavimo na *novaDatabse*, prijavne podatke storitve *novaUser* in *novaDatabase*.

$ mysql -u root -p
mysql> CREATE DATABASE novaDatabase;
mysql> GRANT ALL PRIVILEGES ON novaDatabase. TO 'novaUser'@'<ime_našega_strežnika>' IDENTIFIED BY 'novaDBPass';*
mysql> GRANT ALL PRIVILEGES ON novaDatabase. TO 'novaUser'@'%' *
IDENTIFIED BY 'novaDBPass';
mysql exit

Sinhroniziramo podatkovne baze in tvorimo tabele, ki jih storitev uporablja:

$ sudo nova-manage db sync

 Pri nastavitvi Nove se privzeta podatkovna baza SQLite ne ustvari.

 Pri tem ukazu moramo biti pazljivi, saj je ukaz db sync brez podčrtaja.

Dostop do jedra OS Linuxa običajnim uporabnikom ni omogočen; to onemogoča hipernadzornik, ki upravlja navidezne stroje. S spodnjim ukazom bomo spremenili dostopne pravice do datotek in tako bomo omogočili branje stanja jedra OS-a našega strežnika:

$ sudo dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-$(uname -r)

Vsakič, ko posodobimo naložene programske pakete, se lahko spremenijo tudi pravice dostopa hipernadzornika. Da bi hipernadzornik v bodoče lahko pravilno delal, je potrebno omogočiti dostop tudi po vseh nadaljnjih posodobitvah jedra. Zato je potrebno ustvariti skripto *bash,* s pomočjo ukaza *sudo nano /etc/kernel/postinst.d/statoverride* in napisati del kode, ki bo vsakič, ko se spremenijo dostopne pravice do jedra, omogočila tudi pravico za hipernadzornik.

.
```
#!/bin/sh
version="$1"
# passing the kernel version is required
[ -z "${version}" ] && exit 0
dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-${version}
```

Skripto je potrebno nastaviti kot izvršljivo datoteko:

$ sudo chmod +x /etc/kernel/postinst.d/statoverride

Namestitev in nastavitev komponente za blokovno shrambo

Naslednjo komponento, ki jo namestimo, je Cinder. Komponenta omogoča trajno shranjevanje podatkov virtualnega stroja tudi, če ga uničimo. Uporabili bomo LVM, ki upravlja z bločnimi napravami. Postopek namestitve je podoben kot pri ostalih komponentah. Začnemo z namestitvijo storitve Glance:

$ sudo apt-get install cinder-api cinder-scheduler cinder-volume lvm2

Nastavimo parametre za povezavo v datoteki */etc/cinder/cinder.conf,* dodamo razdelek *[database]*:

connection = mysql://cinderUser:cinderDBPass@<ime_našega_strežnik>/cinderDatabase

 Zapomniti si je potrebno prijavne podatke storitve Cinder za dostop do podatkovne baze.

 Razdelek *[database]* ne obstaja, potrebno ga je napisati.

Dodamo prijavne podatke za identifikacijo storitev v nastavitveno datoteko:

[keystone_authtoken]
auth_uri = <ime_našega_računalnika>:5000
auth_host = <ime_našega_strežnika>
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinderService
admin_password = cinderPass

Cinder za svoje delovanje potrebuje sporočilni strežnik, ki ga je potrebno nastaviti:

[DEFAULT]
...
rpc_backend = rabbit
rabbit_host = <ime_našega_strežnika>
rabbit_port = 5672 //nastavimo vrata, na katerih bosta komunicirala sporočilni strežnik in storitev
rabbit_userid = guest
rabbit_password = rabbitpass

V enaki datoteki pod istim razdelkom nastavimo še gostitelja blokovne shrambe:

glance_host = <ime_našega_računalnika>

Po spreminjanju nastavitev v nastavitveni datoteki je potrebno zagnati procese:

$ sudo service cinder-scheduler restart

$ sudo service cinder-api restart

Ustvarimo podatkovno bazo za storitev Glance in kasneje sinhronizacijo storitve ter baze:

```
$ mysql -u root -p
mysql> CREATE DATABASE cinderDatabase;
mysql> GRANT ALL PRIVILEGES ON cinderDatabase.* TO
'cinderUser'@'<ime_našega_računalnika>'  IDENTIFIED BY 'cinderDBPass';
mysql> GRANT ALL PRIVILEGES ON cinderDatabase.* TO 'cinderUser'@'%' \
        IDENTIFIED BY 'cinderDBPass';
mysql> exit

$ sudo cinder-manage db sync
```