

Sveučilište u Rijeci

Tehnički fakultet

Automatizacija podizanja instanci u OpenStack okruženju i upute za instalaciju *Service Composition* alata

Izradio: Mario Dubravac

Ožujak 2017. Rijeka

Sadržaj

1.	Uvod	3
2.	Automatizirana skripta za podizanje OpenStack instanci	4
3.	Alat za izračun i crtanje grafa normalne distribucije	5
3.1.	Konfiguracija administratorskog računala	5
3.2.	Pokretanje automatizirane skripte za instalaciju	7
3.2.1.	Opis skripte <i>install.yaml</i>	7
3.2.2.	Opis skripti <i>client.yaml</i> , <i>server1.yaml</i> , <i>server2.yaml</i> i <i>server3.yaml</i>	8
3.3.	Ručno podešavanje alata	9
3.3.1.	Stvaranje i konfiguriranje OpenStack instanci	9
3.3.2.	Instalacija potrebnih paketa	11
3.3.3.	Konfiguracija i pokretanje skripti	12

1. Uvod

Svrha ovog seminarskog rada je izraditi alat kojim će se automatizirati postupak podizanja određenog broja instanci unutar OpenStack okruženja koje će sadržavati sve potrebne pakete i testni program koji putem SOAP protokola vrši komunikaciju među servisima na instancama. Pomoću njega, korisniku koji ispred sebe ima podignuto OpenStack okruženje pruža se mogućnost stvaranja odabranog broja instanci, te nakon definiranja koji servis komunicira s kojim te kojem je za rad potrebna baza podataka program automatski generira potrebne heat skripte pomoću kojih se automatizira proces podizanja spomenutih instanci. Na svakoj instanci generira se web servis pisan u php-u koji svojim pokretanjem šalje zahtijev svim servisima s kojim mu je definirana mogućnost komunikacije te kao odgovor dobiva povratnu tekstualnu informaciju da je komunikacija uspješno uspostavljena. Pomoću toga korisnik na vrlo brz i učinkovit način može prilagoditi servise da izvršavaju funkcionalnost za koju su namijenjeni. Cjelina 2 opisuje taj alat. Drugi zadatak je bio opisati postupak izrade alata koji je na sličan način napravljen u OpenStacku unutar SEIP laboratorija, a služi za praćenje zauzeća resursa instanci OpenStacka, ima generator prometa, te sadrži tri instance. Prva instanca na temelju unosa korisnika srednje vrijednosti i standardne devijacije računa parametre grafa normalne distribucije. Zatim te informacije šalje servisu na drugoj instanci koji ih sprema u bazu podataka. Treća instanca uzima te vrijednosti te stvara grafove normalne distribucije koje vraća korisniku kao odgovor na njegove ulazne parametre. U cjelini 3 je detaljno opisan postupak instalacije ovog alata u prazno OpenStack okruženje pokretanjem automatizirane heat skripte i ručno korak-po-korak.

2. Automatizirana skripta za podizanje OpenStack instanci

Računalo s kojeg se skripta pokreće potrebno je imati pristup OpenStack okruženju te imati instaliran Ubuntu operacijski sustav i python programski jezik. Python je od Ubuntu verzije 14.04 preinstaliran na operacijskom sustavu.

Program se pokreće pokretanjem naredbe:

```
python create.py
```

Program prvo pita unos imena ubuntu slike, flavora, imena key-paira te naziva javne i privatne mreže. Moguće je ostaviti default vrijednosti. Nakon toga unosi se broj instanci, veze među instancama i brojevi instanci koji moraju imati mySQL bazu podataka instaliranu. Nakon toga program generira sve potrebne datoteke i sprema ih u direktori *files*. To su jednostavne heat skripte koje sadrže sve potrebne informacije za podizanje instanci, te instalaciju potrebnih servisa pisanih u php programskom jeziku. Direktorij *templates* sadrži općenite uzorke pojedinih heat skripti na temelju kojih se izražuju potrebne skripte. Nakon toga potrebno je samo pokrenuti heat skriptu *install.yaml* i ona će obaviti sav posao. Na tom računalu potrebno je imati instaliran heat klijent.

```
apt-get install python-heatclient
```

Pokretanjem skripte *install.yaml* izvrši se stvaranje svih potrebnih instanci i instaliraju svi potrebni paketi i time je sustav spreman za izvođenje. Skripta se pokreće iz Horizon sučelja ili iz komandne linije naredbom:

```
heat stack-create ime_stacka -f install.yaml
```

Instancama se pristupa putem naredbe:

```
ssh -i <lokacija_kljuca.pem> ubuntu@<floating_ip_client>
```

Nakon logiranja putem *ssh* protokola na neku od instanci potrebno je pokrenuti određenu php skriptu koja se nalazi u */var/www/html/* direktoriju pomoću naredbe:

```
php skripta<broj_instance>.php
```

Nakon toga u komandnoj liniji vidljiv je ispis s kojim servisima pokrenuta skripta komunicira.

3. Alat za izračun i crtanje grafa normalne distribucije

Instalacija alata za kompoziciju servisa sastoji se od konfiguracije jednog računala na kojem će se vrtiti generator prometa i skripta za praćenje zauzeća resursa (u daljnjem tekstu administratorsko računalo) i četiri OpenStack instance. Preduvjet za instalaciju alata je podignuto kompletno OpenStack okruženje. Važno je napomenuti da skripte neće raditi u DevStack okruženju. Također, potrebno je potpuno proći OpenStack vježbe za kolegij *Upravljanje u programskom inženjerstvu* autora Nikole Domazeta. U poglavlju 1.1. opisano je konfiguriranje administratorskog računala. Za administratorsko računalo potrebno je da se nalazi unutar mreže OpenStacka. U slučaju rada unutar SEIP laboratorija dovoljno je priključiti računalo putem LAN-a na mrežu ili koristiti jedno od računala u SEIP laboratoriju. Za sad nije moguće raditi sa računala izvan mreže OpenStacka jer u tom slučaju računalo nije u mogućnosti ostvariti vezu sa OpenStack instancama. U poglavljima 1.2. i 1.3. opisane su dvije metode instalacije alata. Prva metoda je pokretanje skripte koja automatizira proces stvaranja instanci i instalacije potrebnih paketa i skripti, a druga metoda opisuje ručnu instalaciju korak po korak. Unutar OpenStacka potrebno je imati četiri instance od kojih je jedna instanca namijenjena za spajanje klijenata i pokretanje samog programa, a na ostale tri se nalaze web servisi koji sačinjavaju aplikaciju.

3.1. Konfiguracija administratorskog računala

Administratorsko računalo mora imati instaliran Ubuntu operacijski sustav i najnovija ažuriranja sustava. U daljnjem tekstu prikazani su svi paketi koje administratorsko računalo mora imati instalirane.

```
apt-get update
apt-get -y upgrade
apt-get -y install git
apt-get install build-essential python-dev**
apt-get install python-pip
apt-get install python-heatclient
pip install pymongo
pip install flask
pip install pecan
pip install wsme
pip install python-ceilometerclient
pip install python-keystoneclient
pip install python-openstackclient
pip install netifaces
```

Nakon instalacije paketa, potrebno je povući skripte za generator prometa i praćenje zauzeća resursa. To se vrši slijedećim naredbama.

```
wget --no-check-certificate
https://raw.githubusercontent.com/lpejak/ServiceComposition/master/python
/ceilo_monitoring.py

wget --no-check-certificate
https://raw.githubusercontent.com/lpejak/ServiceComposition/master/python
/thread.py

wget --no-check-certificate
https://raw.githubusercontent.com/lpejak/ServiceComposition/master/python
/credentials.py
```

Te skripte se pokreću naredbom:

```
python <naziv_skripte>.py
```

Lokacija skripti na računalu nije bitna. Dalje je potrebno napraviti novu OS sliku koja će se zvati UbuntuCloud14, a njen izvor je <https://cloud-images.ubuntu.com/trusty/current/trusty-server-cloudimg-amd64-disk1.img>. Ona je korištena i u OpenStack vježbama za kolegij *Upravljanje u programskom inženjerstvu* te je tamo opisan postupak izrade specifične OS slike. Također, potrebno je stvoriti novi key pair koji će se pridruživati instancama za pristup njihovoj konzoli putem SSH protokola i pohraniti ga na računalu.

Na administratorskoj strani potrebno je još imati downloadan OpenStack RC File v2.0 te pomoću naredbe source dodijeliti vrijednosti environment varijablama.

```
source <naziv_skripte>.sh
```

Još je potrebno promijeniti id-eve instanci unutar `thread.py` i `ceilo_monitoring.py` da odgovaraju id-evima nakon što se kreiraju instance. Unutar skripte `ceilo_monitoring.py` potrebno je dodati početno i krajnje vrijeme te datum za kojeg pratimo zauzeće resursa. To se vrši postavljanjem varijabli `hourStart`, `hourEnd`, `minuteStart`, `minuteEnd`, `year`, `month` i `day`. Varijable `hourStart` i `hourEnd` primaju vrijednosti od 0 do 24, `minuteStart` i `minuteEnd` od 0 do 50 (za više od 50 program ionako uzima vrijednost 50), `year` pripadnu godinu, `month` mjesec od 1 do 12 i `day` dan u mjesecu.

3.2. Pokretanje automatizirane skripte za instalaciju

Nakon što je administratorsko računalo podešeno potrebno je povući skriptu `install.yaml` u radno okruženje Ubuntu sustava.

```
wget --no-check-certificate  
https://raw.githubusercontent.com/lpejak/ServiceComposition/master/heat/i  
ninstall.yaml
```

Nakon toga treba samo provjeriti jesu li parametri u skripti točno upisani (`image`, `key`, `private_network`, `public_network`). Ukoliko nisu, potrebno ih je korigirati i spremati promjene.

Pokretanjem te skripte izvrši se stvaranje svih potrebnih instanci i instaliraju svi potrebni paketi i time je sustav spreman za izvođenje. Skripta se pokreće iz Horizon sučelja ili iz komandne linije naredbom:

```
heat stack-create ime_stacka -f install.yaml
```

Prosječno vrijeme za izvršavanje skripte je oko četiri sata. Instancama se pristupa putem naredbe:

```
ssh -i <lokacija_kljuca.pem> ubuntu@<floating_ip_client>
```

Nakon ovog koraka alat bi trebao biti funkcionalan. Njemu se pristupa pokretanjem `index.html` putem internet preglednika (`<ip_adresa_klijenta>/index.html`) i unosom parametara normalne distribucije ili direktnim pokretanjem `client.php` pri čemu se parametri generiraju pseudoslučajno. To se pokreće unosom `IP` adrese u web preglednik, odnosno unosom `<ip_adresa_klijenta>/client.php`. Na administratorskoj strani je moguće pokrenuti `thread.py` čime se automatski generira promet i pokreću skripte po po predefiniranoj distribuciji, a pokreće se naredbom `python thread.py`.

3.2.1. Opis skripte *install.yaml*

Skripta `install.yaml` služi za samu instalaciju alata u postojeće OpenStack okruženje. U njoj se automatski pozivaju i izvršavaju skripte `client.yaml`, `server1.yaml`, `server2.yaml` i `server3.yaml` koje služe za podizanje instanci i instalaciju potrebnih paketa na instance. Skripta se nalazi na slijedećoj adresi:

<https://github.com/lpejak/ServiceComposition/blob/master/heat/install.yaml>.

Polje *parameters* sadrži parametre koje će imati svaka izrađena instanca, a to su redom:

- *image* – naziv OS slike na kojima će se bazirati instance
- *flavor* – tip flavora na kojima će se bazirati instance, poželjno da bude *m1.small*
- *key* – naziv para ključa pomoću kojeg će se kasnije pristupati instancama
- *private_network* – naziv privatne mreže unutar koje se stvaraju instance
- *public_network* – naziv javne mreže na koju su instance spojene

Polje *resources* definira OpenStack resurse koje skripta stvara. To su *Security Group* te četiri navedene instance. U *Security Group* su dodani *TCP* portovi 22 za *SSH* i 80 za *HTTP* te *ICMP* protokol za *ping*.

Resursi *client*, *server1*, *server2* i *server3* stvaraju se na isti način, jedino je razlika u tome što *server1*, *server2* i *server3* vraćaju *IP* adrese prilikom kreiranja koje se na kraju zapisuju u datoteku *serviceIP.txt* na klijentskoj strani te se od tuda koriste za SOAP komunikaciju.

3.2.2. Opis skripti *client.yaml*, *server1.yaml*, *server2.yaml* i *server3.yaml*

Skripte *client.yaml*, *server1.yaml*, *server2.yaml* i *server3.yaml* ne pozivaju se direktno svaka posebno nego se automatski pozivaju iz *install.yaml* skripte prilikom njenog pokretanja.

Skripte se nalaze u slijedećem linku <https://github.com/lpejak/ServiceComposition/tree/master/heat>.

Napisane su na isti način, jedino se razlikuju u kodu koji se izvršava na instancama nakon što se one kreiraju te *client.yaml* prima tri parametra više, *IP* adrese svakog servera. Ostali parametri (*image*, *flavor*, *key*, *private_network*, *public_network*, *security_group*) dobivaju se iz *install.yaml* skripte.

U svakoj skripti unutar *template* odjeljka nalazi se *bash* skripta koja izvršava instalaciju svih paketa. Vidljivo je da *IP* adrese skripta na klijentskoj strani pohranjuje u *serviceIP.txt* datoteku.

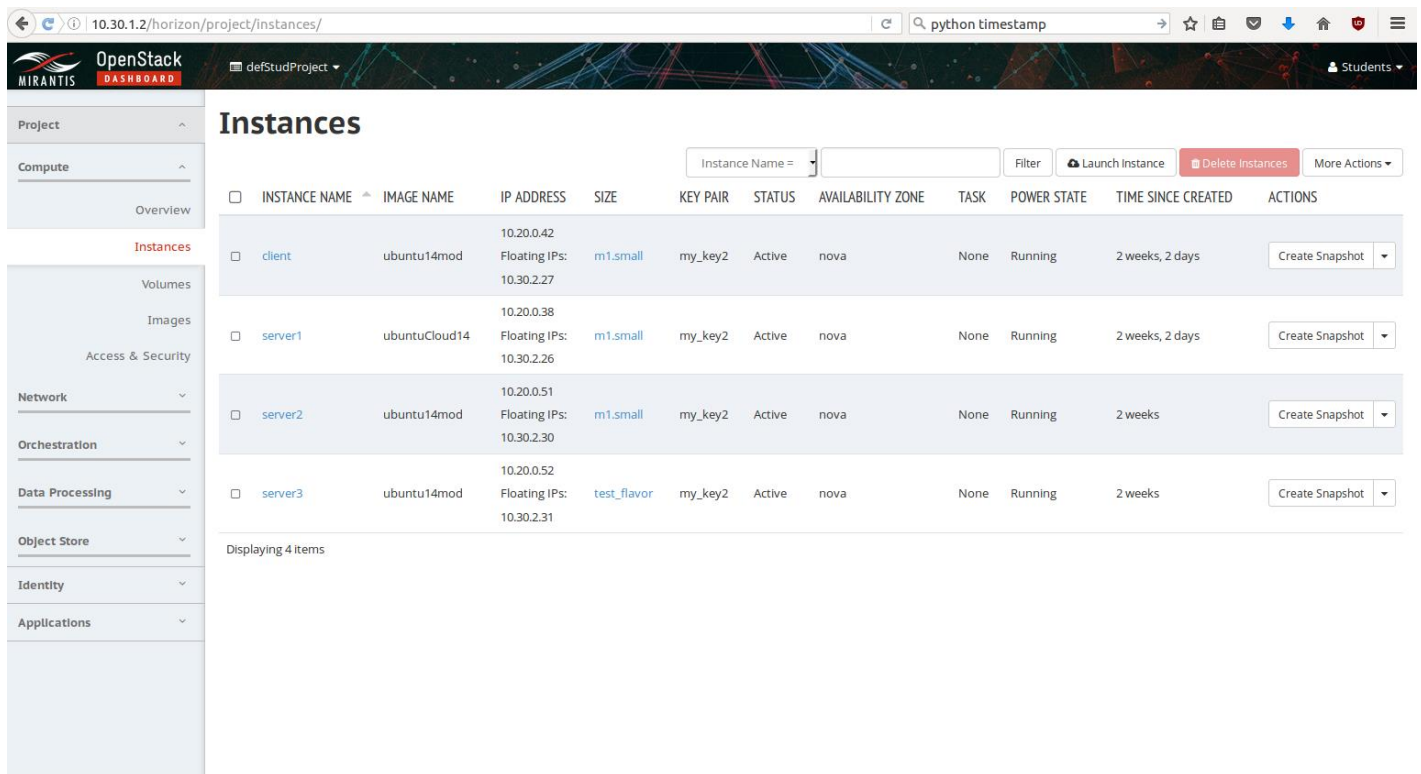
3.3. Ručno podešavanje alata

Instalacija i podešavanje alata ručnom metodom sastoji se od 3 koraka:

1. Stvaranje i konfiguriranje OpenStack instanci
2. Instalacija potrebnih paketa na pojedinoj instanci
3. Konfiguracija i pokretanje skripti

3.3.1. Stvaranje i konfiguriranje OpenStack instanci

Iako je moguće svaku instancu konfigurirati pojedinačno, s obzirom da sve instance velik dio paketa imaju jednak preporuča se stvoriti jednu instancu, iskonfigurirati ju, unutar OpenStack Horizon sučelja stvoriti snapshot te instance i na temelju dobivenog snapshota kreirati još 3 takve instance te njih dalje konfigurirati pojedinačno. Za ispravan rad instanci potrebno je svakoj dodijeliti najmanje 1GB RAM memorije, 20GB prostora i 1 VCPU stoga je potrebno stvoriti novi flavor s tim resursima. Preporuča se uzeti m1.small flavor koji koristi 2GB RAM memorije ali u tom slučaju smo unutar SEIP laboratorija ograničeni na četiri instance za projekt s obzirom da je ograničenje memorije na 8GB. Za prvu instancu na temelju koje će se kasnije stvoriti snapshot za iduće instance potrebno je kao ulaznu sliku uzeti ubuntu sliku (<https://cloud-images.ubuntu.com/trusty/current/trusty-server-cloudimg-amd64-disk1.img>) koja je korištena i u OpenStack vježbama za kolegij *Upravljanje u programskoj inženjerstvu* te je tamo opisan postupak izrade specifične OS slike. Također, instanci je potrebno tijekom stvaranja pridružiti key pair i nakon stvaranja pridružiti floating IP adresu. Naknadno, potrebno je u Security Group dodati dozvole za *ping* i *ssh* kao što je opisano u prethodno spomenutim vježbama.



Slika 1. Instance kreirane u OpenStack Horizon okruženju

Na slici 1 prikazan je primjer kreiranih instanci u OpenStack Horizon sučelju.

U daljnjem tekstu opisan je postupak konfiguracije prve instance. Neka se ona zove *client*, a kada se do kraja podesi na temelju nje se pod akcijom *Create Snapshot* spremi trenutno stanje slike pod novom slikom. Na temelju te nove slike stvorimo još 3 instance (*server1*, *server2* i *server3*) na kojima se izvršavaju web servisi.

S obzirom da su *client* instanca i administratorsko računalo s preinstaliranim Ubuntu operacijskim sustavom unutar iste OpenStack mreže, oni moraju međusobno komunicirati. Njihovu komunikaciju moguće je provjeriti naredbom *ping*.

```
ping <floating_ip_client>
```

Pomoću SSH protokola ulazimo u sučelje klijenta sa administratorskog računala pomoću naredbe:

```
ssh -i <lokacija_ključa.pem> ubuntu@<floating_ip_client>
```

Nakon toga trebali bi biti spojeni na komandnu liniju klijenta. Zatim je potrebno promijeniti lozinku root korisnika slijedećom naredbom te podesiti lozinku koju želimo. Potom je moguće spajanje preko ssh protokola bez korištenja ključa.

```
sudo passwd
```

Nakon postavljanja ovlasti potrebno je izvršiti slijedeće naredbe za instalaciju potrebnih paketa:

```
apt-get update
apt-get -y upgrade
sudo debconf-set-selections <<< 'lamp-server^ mysql-
server/root_password password grad'
sudo debconf-set-selections <<< 'lamp-server^ mysql-
server/root_password_again password grad'
apt-get -y install git
apt-get -y install lamp-server^
```

Time se softver ažurira na najnoviju verziju, instalira git i lamp server.

Nakon što je sve navedeno instalirano potrebno je napraviti snapshot klijenta i na bazi tog snapshota 3 instance.

3.3.2. Instalacija potrebnih paketa

Nakon što su instance uspješno pokrenute, potrebno je preko SSH protokola pristupiti svakoj pojedinačno te kopirati odgovarajuće skripte i instalirati pakete koji su potrebni za pojedinu instancu.

Sve php i html skripte na instancama moraju nalaziti točno unutar direktorija `/var/www/html`. Klijent mora imati datoteke `index.html`, `client.php` i `nusoap.php`. Na serveru 1 moraju se nalaziti datoteke `service.php` i `nusoap.php`, a na serveru 2 `mysql.php` i `nusoap.php`. MySQL-u se pristupa pomoću slijedeće naredbe,

```
mysql -uroot -pgrad
```

gdje je korisničko ime root, a lozinka grad.

Server 3 mora imati skripte rbg.inc.php, graph.php, nusoap.php, phplot.php i treba imati dodane ovlasti za read i write nad direktorijem.

```
chmod a+rw <ime_direktorija>
```

Još je potrebno instalirati gd knjižnicu i restartati apache server.

```
apt-get install php5-gd && service apache2 restart
```

Nakon ovog koraka sve bi trebalo biti spremno za korištenje.

3.3.3. Konfiguracija i pokretanje skripti

Prije nego se aplikacija pokrene još je potrebno prilagoditi određene varijable da odgovaraju trenutnom projektu. Unutar client.php skripte na klijentskoj strani potrebno je unijeti točne floating IP adrese u varijable \$servis1, \$servis2 i \$servis3. One se mogu saznati iz Horizon sučelja kao što je prikazano na slici 1. Unutar skripte ceilo_monitoring.py potrebno je dodati početno i krajnje vrijeme i datum za kojeg pratimo zauzeće resursa. To se čini postavljanjem varijabli hourStart, hourEnd, minuteStart, minuteEnd, year, month i day. Varijable hourStart i hourEnd primaju vrijednosti od 0 do 24, minuteStart i minuteEnd od 0 do 50 (za više od 50 program ionako uzima vrijednost 50), year pripadnu godinu, month mjesec od 1 do 12 i day dan u mjesecu. Unutar Horizon sučelja potrebno je dodati vrijednosti id-eva instanci i upisati ih u polja id1, id2 i id3. Nakon ovog koraka alat bi trebao biti funkcionalan. Njemu se pristupa pokretanjem index.html na klijentskoj strani i unosom parametara normalne distribucije ili direktnim pokretanjem client.php pri čemu se parametri generiraju pseudoslučajno. To se pokreće unosom IP adrese u web preglednik, odnosno unosom <ip_adresa>/client.php. Na administratorskoj strani je moguće pokrenuti thread.py čime se automatski generira promet i pokreću skripte po predefiniciji distribuciji, a pokreće se naredbom `python thread.py`.