

Corso di calcolo distribuito e sistemi ad alte prestazioni



Relazione: Cloud Computing

De Angelis Fabio
matricola: 318037

Introduzione

In questa relazione si descrive la realizzazione del progetto relativo al Cloud Computing realizzato nell'ambito del corso di calcolo distribuito e sistemi ad alte prestazioni.

Il progetto prevede la realizzazione di un ambiente cloud privato tramite Openstack.

Con il termine Cloud Computing si indica un paradigma di erogazione di servizi offerti on demand da un fornitore ad un cliente finale attraverso la rete Internet (come l'archiviazione, l'elaborazione o la trasmissione dati), a partire da un insieme di risorse preesistenti, configurabili e disponibili in remoto sotto forma di architettura distribuita.

Le risorse non vengono pienamente configurate e messe in opera dal fornitore appositamente per l'utente, ma gli sono assegnate, rapidamente e convenientemente, grazie a procedure automatizzate, a partire da un insieme di risorse condivise con altri utenti lasciando all'utente parte dell'onere della configurazione.

Il principio alla base del cloud è quindi quello di offrire potenza di calcolo, memoria e software come servizi.

Openstack è un cloud operative system in grado di gestire un insieme di risorse di vario tipo (computazionali, di networking e di memoria) andando a creare un data center che viene gestito attraverso una dashboard che fornisce numerosi strumenti di controllo all'amministratore.

Architettura del sistema

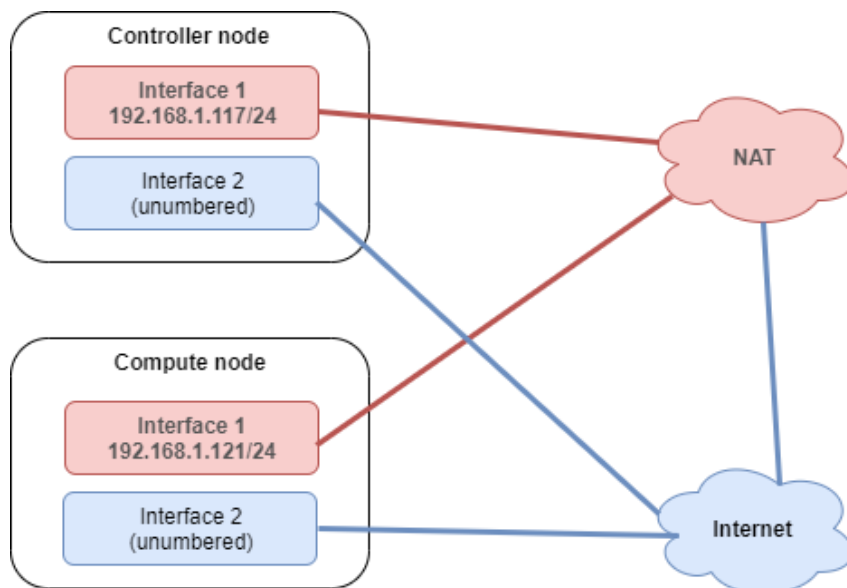
Prima di procedere all'installazione di Openstack è necessario definire e implementare l'architettura alla base del sistema.

In questo caso è stato implementato un sistema costituito da una macchina con installato il sistema operativo Ubuntu 16.04 e da una macchina virtuale con sistema operativo Lubuntu.

Entrambe le macchine hanno due interfacce, la prima con indirizzo ip statico serve a collegarle tra loro, l'altra con indirizzo ip dinamico serve a collegarle ad Internet.

La macchina fisica va a costituire il cosiddetto "controller node" ossia il nodo amministratore, mentre la macchina virtuale costituisce il "compute node", questi due nodi vanno a formare la configurazione minima necessaria all'implementazione di un cloud privato.

È quindi necessario configurare le interfacce di rete e gli host names, nello specifico al controller node è stato assegnato indirizzo ip statico 192.168.1.117/24 mentre al compute node l'indirizzo 192.168.1.121/24, ovviamente a entrambi gli indirizzi ip è stato assegnato il corrispondente host name (controller e compute) nell'apposito file di configurazione (/etc/hosts) su entrambe le macchine.



Procedura d'installazione

1) Controller node



Per l'installazione di Openstack sul controller node è stato utilizzato Devstack.

Devstack è un insieme di scripts che consente l'installazione di Openstack in alternativa all'utilizzo di cloud pubblici (che sono a pagamento), andando quindi a fornire un metodo di installazione semplice e completamente gratuito.

Per prima cosa è stato creato un nuovo utente di nome stack a cui sono stati assegnati privilegi sudo.

Una volta effettuato l'accesso tramite utente stack, è stata scaricata la repository di devstack tramite git e all'interno della directory corrispondente è stato creato un file di configurazione "local.conf" in cui sono state definite tutte le password necessarie ossia ADMIN_PASSWORD, RABBIT_PASSWORD, SERVICE_PASSWORD E DATABASE_PASSWORD.

Definite le password si procede lanciando in esecuzione lo script **stack.sh**, che dà il via alla procedura d'installazione

Stack.sh andrà ad installare tutti i componenti software necessari al funzionamento dell'ambiente cloud, ossia:

- **mariaDB**, (qualora non sia già installato) ossia il database management system necessario alla creazione e alla gestione del database
- **rabbitmq**, ossia un message queue service, costituito da un broker di messaggi, ovvero un programma intermedio che traduce un messaggio dal protocollo di messaggistica formale del mittente al protocollo di messaggistica formale del ricevitore
- **memcache**, servizio che ha il compito di memorizzare nella cache i tokens utilizzati dall'identity service per l'autenticazione.
- **Keystone**, l'identity service di Openstack, ovvero un servizio che permette la gestione delle autenticazioni, delle autorizzazioni e del catalogo dei servizi. L'identity service è il primo servizio con cui un utente interagisce e, una volta autenticato, l'end user può utilizzare la propria identità per accedere ai servizi Openstack a lui disponibili (in base al suo ruolo).
- **Glance**: l'image service di Openstack, ossia il servizio che permette agli utenti di trovare, registrare e richiamare le immagini delle macchine virtuali, ciò è possibile attraverso una REST API che permette di interrogare i metadati delle macchine virtuali per ottenerne l'immagine. Questo servizio è di centrale importanza nei sistemi IaaS (Infrastructure as a Service).
- **Nova** : il compute service di Openstack, articolato in vari moduli, ha sostanzialmente il compito di gestire le API e istanze tra i vari utenti e i vari nodi che compongono il sistema.
- **Neutron**: il networking service, che permette la gestione della rete e delle sue singole interfacce permettendone l'aggiunta, la rimozione e la modifica.
- **Cinder**: il block storage service, ossia il servizio che permette la gestione della memoria del sistema e delle singole istanze.
- **Dashboard**: la dashboard (pannello di controllo) di Openstack che fornisce un'interfaccia grafica intuitiva per il controllo del sistema dal controller node

```

=====
DevStack Component Timing
(times are in seconds)
=====
run_process      17
test_with_retry   2
apt-get-update    4
pip_install      280
osc              109
wait_for_service  11
git_lined        224
dbsync           24
apt-get           67
-----
Unaccounted time  317
-----
Total runtime     1055

This is your host IP address: 192.168.1.117
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.1.117/dashboard
Keystone is serving at http://192.168.1.117/identity/
The default users are: admin and demo
The password: Fab1o1995

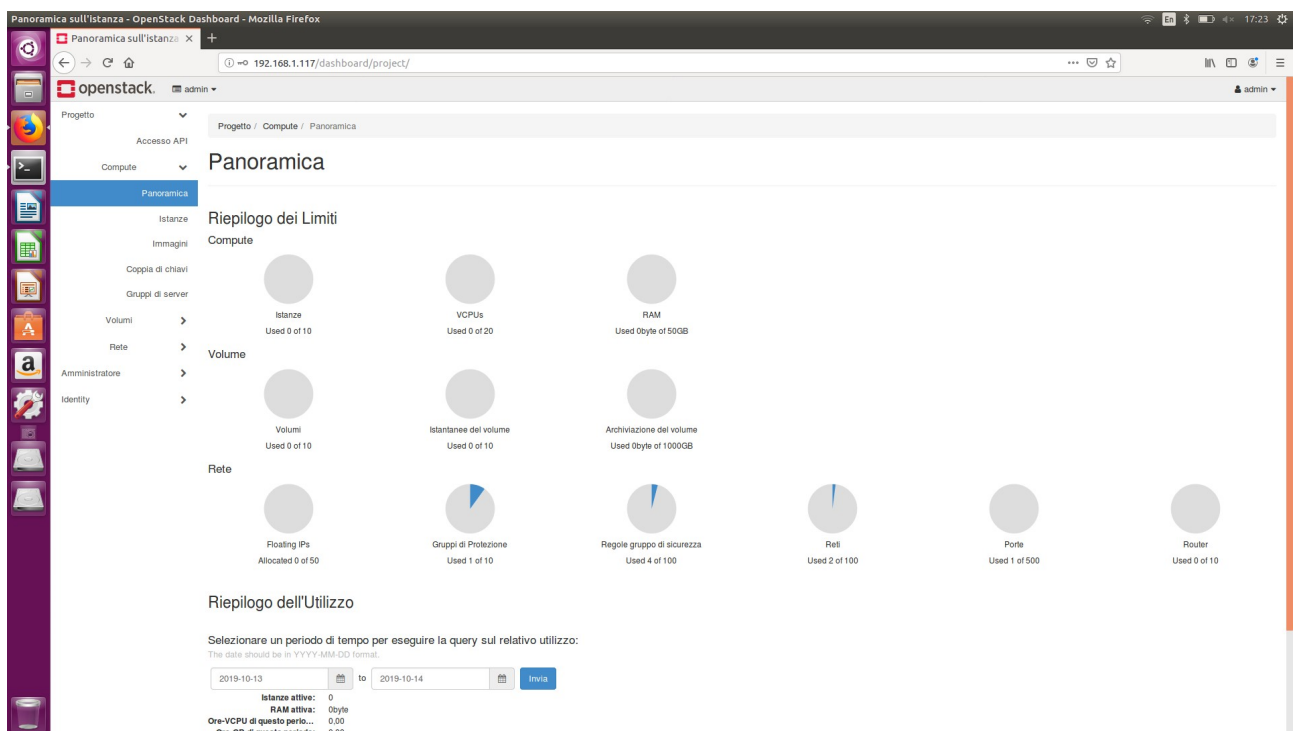
WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future

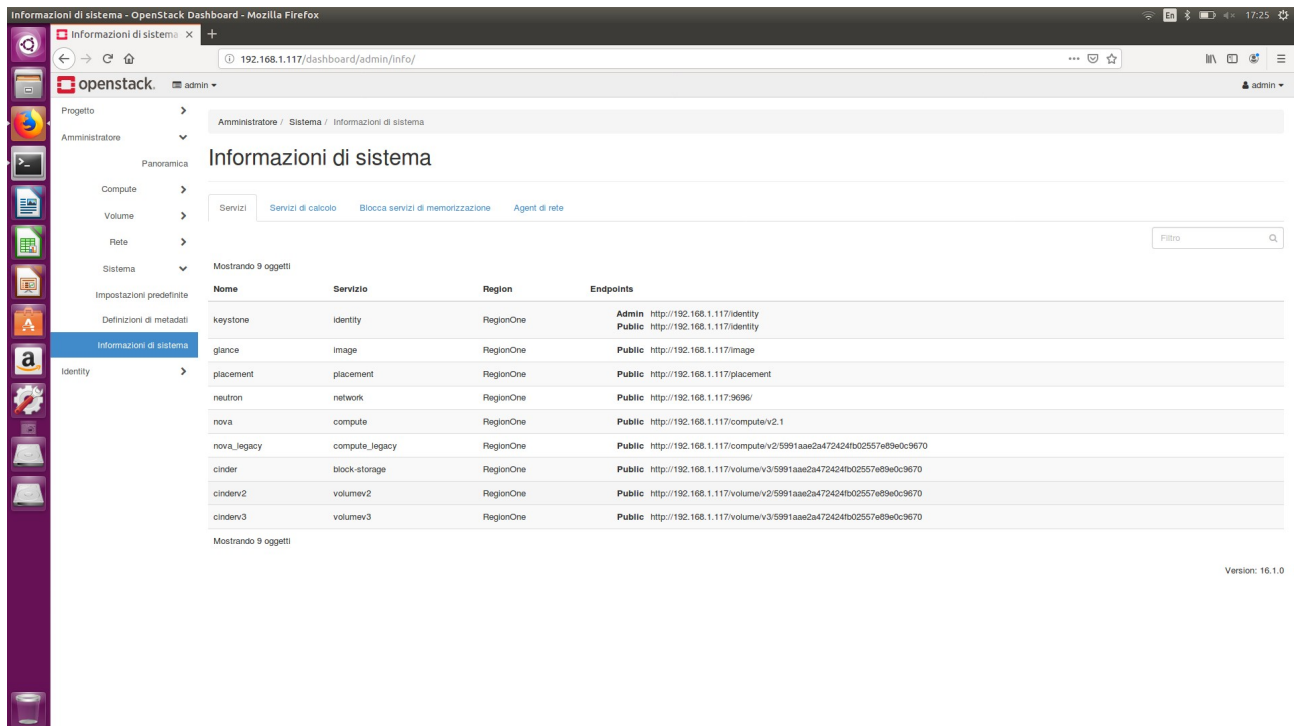
Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: ussur1
Change: 7840b6e291b5e28ba96ce871b5e01d66c14d9201 Merge "Ensure that the zuul home can be traversed" 2019-10-08 19:33:53 +0000
OS Version: Ubuntu 16.04 xenial

```

Conclusa la procedura d'installazione, apparirà la schermata visibile nell'immagine precedente e si potrà accedere alla dashboard di Openstack da localhost, ossia all'indirizzo 192.168.1.117, effettuando il login tramite le credenziali dell'amministratore definite in precedenza.





2) Compute Node

Per quanto riguarda il compute node la procedura di installazione dei servizi e di aggiunta al network, è ovviamente diversa.

Una volta definito hostname ed indirizzo della macchina su entrambi i nodi, si procede andando ad installare il compute service Nova sul compute node.

Per farlo è sufficiente eseguire il comando:

```
apt install nova
```

Terminata la procedura d'installazione bisogna configurare il compute service, e per farlo si va a modificare il relativo file di configurazione: /etc/nova/nova.conf.

All'interno di nova.conf sono stati modificati/aggiunti vari campi:

- All'interno della sezione [DEFAULT]:all'interno della sezione: [api]:

```
GNU nano 2.5.3      File: /etc/nova/nova.conf

[api]
#
# Options under this group are used to define Nova API.
#
# From nova.conf
#
#
# This determines the strategy to use for authentication: keystone or noauth2.
# 'noauth2' is designed for testing only, as it does no actual credential
# checking. 'noauth2' provides administrative credentials only if 'admin' is
# specified as the username.
# (string value)
# Possible values:
# keystone - <No description provided>
# noauth2 - <No description provided>

auth_strategy = keystone
```

- all'interno della sezione [keystone_authtoken]:

```
auth_url = http://controller:5000/v3
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = Fabio1995
```

- all'interno della sezione [vnc]:

```
#my configuration
enabled = True :
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://controller:6080/vnc_auto.html
```

- all'interno della sezione [glance]:

```
#
# NOTE: The preferred mechanism for endpoint discovery is via keystoneauth1
# loading options. Only use api_servers if you need multiple endpoints and are
# unable to use a load balancer for some reason.
#
# Possible values:
#
# * A list of any fully qualified url of the form
# "scheme://hostname:port[/path]"
# (i.e. "http://10.0.1.0:9292" or "https://my.glance.server/image").
# (list value)

api_servers = http://controller:9292
```

- all'interno della sezione [oslo_concurrency]

```
#
# From oslo.concurrency
#
# Enables or disables inter-process locks. (boolean value)
#disable_process_locking = false
#
# Directory to use for lock files. For security, the specified directory should
# only be writable by the user running the processes that need locking. Defaults
# to environment variable OSLO_LOCK_PATH. If OSLO_LOCK_PATH is not set in the
# environment, use the Python tempfile.gettempdir function to find a suitable
# location. If external locks are used, a lock path must be set. (string value)

lock_path = /var/lib/nova/tmp
```

Terminata la configurazione si riavvia il servizio per applicare le modifiche, tramite:
service nova-compute restart

Terminata l'installazione e la configurazione del compute service Nova si è passati all'installazione del networking service Neutron.

Innanzitutto si è installato il pacchetto da terminale tramite comando:

```
apt install neutron-linuxbridge-agent
```

Una volta conclusa la procedura d'installazione, anche in questo caso, è necessario configurare il servizio, in questo caso si va a modificare il file di configurazione /etc/neutron/neutron.conf:

- nella sezione [database] si va a rimuovere qualsiasi opzione di connessione già definita poiché il compute node per definizione deve avere accesso al database.

- Nella sezione [default]:

```
# The type of authentication to use (string value)
auth_strategy = keystone

#
# driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query
#
# Example: rabbit://rabbitmq:password@127.0.0.1:5672//
#
# For full details on the fields in the URL see the documentation of
# oslo_messaging.TransportURL at
# https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
# (string value)
transport_url = rabbit://openstack:Fabio1995@controller
```

- nella sezione [keystone_authtoken]:

```
auth_uri = http://controller:5000
auth_url = http://controller:3557

# Optionally specify a list of memcached server(s) to use for caching. If left
# undefined, tokens will instead be cached in-process. (list value)
# Deprecated group/name - [keystone_authtoken]/memcache_servers
# Authentication type to load (string value)
# Deprecated group/name - [keystone_authtoken]/auth_plugin
auth_type = password

# Config Section from which to load plugin specific options (string value)
#auth_section = <None>

project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = Fabio1995
```

Non basta però editare solo neutron.conf per completare la configurazione del servizio ma è necessario editare la sezione [neutron] all'interno del file di configurazione di nova (nova.conf):

```
url = http://controller:9696

# Authentication type to load (string value)
# Deprecated group/name - [neutron]/auth_plugin
auth_type = password

# Config Section from which to load plugin specific options (string value)
#auth_section = <None>

# Authentication URL (string value)
auth_url = http://controller:35357

# Username (string value)
# Deprecated group/name - [neutron]/user_name
username = neutron

# User's domain id (string value)
#user_domain_id = <None>

# User's domain name (string value)
#user_domain_name = <None>

# User's password (string value)
password = Fabio1995
```


Infine, si riavviano entrambi servizi per applicare le modifiche tramite comandi:

```
service nova-compute restart
```

```
service neutron-linuxbridge-agent restart
```

Testing

Una volta completata la procedura d'installazione su entrambi i nodi è stato effettuato un testing per verificare il corretto funzionamento del sistema in ogni suo componente e servizio.

Dal compute node sono stati eseguiti gli appositi comandi per mostrare la lista dei servizi compute e delle immagini, come visibile nell'immagine seguente.

```
stack@fabdea-TM1701:~/devstack$ openstack compute service list
```

ID	Binary	Host	Zone	Status	State	Updated At
3	nova-scheduler	fabdea-TM1701	internal	enabled	up	2019-11-04T18:44:52.000000
6	nova-conductor	fabdea-TM1701	internal	enabled	up	2019-11-04T18:44:52.000000
1	nova-conductor	fabdea-TM1701	internal	enabled	up	2019-11-04T18:44:51.000000
2	nova-compute	fabdea-TM1701	nova	enabled	up	2019-11-04T18:44:45.000000

```
stack@fabdea-TM1701:~/devstack$ openstack image list
```

ID	Name	Status
a5bdefcb-3e1d-4baf-820a-f3c859508af4	cirros-0.4.0-x86_64-disk	active

Infine è stato eseguito un job basilare su entrambi i nodi sempre a scopo di testing:

```
stack@fabdea-TM1701:~/devstack$ ./test
HELLO WORLD!
stack@fabdea-TM1701:~/devstack$
```

Problematiche incontrate

Le principali problematiche incontrate durante l'esperienza si sono riscontrate praticamente solo nella fase d'installazione di Openstack tramite script stack.sh sul controller node. Essendo infatti quest'ultimo uno script pensato per poter essere eseguito su praticamente qualsiasi macchina con una qualsiasi distribuzione linux, durante la sua esecuzione possono nascere varie problematiche legate ad esempio ai permessi di sistema e ai problemi di compatibilità che possono occorrere a cause delle versioni dei software installati o da installare.

In questo caso sono stati riscontrati entrambi i problemi, nello specifico la procedura si è interrotta a causa della mancata presenza della versione di pip python richiesta dallo script e dalla mancanza di permessi sufficienti per l'utente stack.

Problemi che sono stati risolti eseguendo i seguenti comandi da terminale:

```
sudo apt-get install git python-pip
```

```
sudo pip install -upgrade
```

```
sudo virtualenv /home/stack/requirements/.venv/
```

```
sudo chown -R stack /home/stack/.cache
```


Conclusioni

In questa relazione è stato mostrato come implementare un ambiente cloud privato Openstack, strutturato in due nodi: un nodo controller e un nodo compute.

Il nodo controller costituisce il nodo principale che va ad eseguire la maggior parte dei servizi come ad esempio l'identity service, l'image service, la dashboard openstack, il DBMS e il proprio networking agent. Esso va quindi a costituire praticamente il nodo che consente la gestione ed il controllo di tutto il sistema da parte dell'amministratore.

Il compute node, invece, esegue il proprio hypervisor per la gestione delle istanze e il proprio networking agent che connette le istanze alla rete virtuale ed è in grado di fornire servizi di firewalling.

Ovviamente questo sistema fornisce tutti i vantaggi del Cloud Computing, ossia:

- riduzione dei costi
- accesso universale
- flessibilità
- aggiornamenti software
- elevata quantità di spazio di archiviazione a disposizione
- elevata scalabilità
- Quality of Services massima