

Progetto di Sistemi Middleware

Installazione di un sistema distribuito Cloud tramite la piattaforma OpenNebula

Andrea Sghedoni¹,

¹ Università degli Studi di Bologna, Maggio-Giugno 2015, A.A. 2014/2015
MATR. 0000736038, andre.sghedoni4@studio.unibo.it

Abstract. Il progetto consiste nell'installazione e configurazione di un piccolo sistema cloud con gli strumenti offerti da OpenNebula. Il seguente documento ha lo scopo di spiegare i procedimenti che sono stati necessari adottare per mettere in piedi un'architettura del genere e di verificarne alcune peculiarità.

Keywords: Cloud, Sistemi distribuiti, OpenNebula, virtualizzazione

1 INTRODUZIONE

OpenNebula è un'applicazione open source che permette la realizzazione di infrastrutture Cloud distribuite, fornendo tutti gli aspetti principali di queste architetture.

Viene principalmente utilizzato per creare sistemi IaaS (Infrastructure as a Service), dove le risorse hardware e virtuali vengono richieste ed utilizzate in remoto dagli host che si interfacciano al Cloud.

Nell'esercitazione effettuata sono state utilizzate delle immagini di CentOS per la creazione dei nodi coinvolti nel sistema, in particolare il FrontEnd ed il Worker Node.

Queste vengono fatti girare virtualizzate con VMWare, creando gli host sui quali andranno in esecuzione le “virtual machine”, le quali potranno essere attivate e gestite direttamente dall'interfaccia di OpenNebula Sunstone.

Nella nostra architettura il FrontEnd è come un Worker Node, ma designato per erogare i servizi forniti da OpenNebula.

Uno dei principali obiettivi dell'esperienza è verificare la proprietà di Migrazione Live, dove una macchina virtuale si sposta da una risorsa ad un'altra senza stoppare l'esecuzione dei task correnti.

L'Hypervisor KVM è una soluzione per l'utilizzo della virtualizzazione per SO basati su Linux e hardware contenente l'estensione per virtualizzazione (Intel VT o AMD-V).

Infine per la condivisione delle risorse viene utilizzato il NFS, che è un filesystem distribuito tra diversi host in una rete.

In particolare una directory del FrontEnd verrà utilizzata come punto di condivisione risorse con il Worker Node.

2 INSTALLAZIONE INIZIALE E PACCHETTI

La prima azione da svolgere è sicuramente controllare che la macchina su cui si intende lavorare supporti la virtualizzazione.

Ciò può essere fatto con il comando terminale:

```
$ egrep '(vmx|svm)' /proc/cpuinfo
```

Se l'output del comando restituisce dei risultati allora la risposta è affermativa, in caso contrario non sarebbe possibile mettere in piedi l'infrastruttura pensata.

Ora vediamo come installare i componenti per la realizzazione dell'architettura, i seguenti comandi fanno riferimento al SO Ubuntu 12.04 e ad OpenNebula 4.4. Tutti i passaggi sono comunque ben documentati nelle guide ufficiali di OpenNebula (vedere la bibliografia per i riferimenti considerati).

Detto ciò, di seguito sono mostrati i passi per l'installazione di tutti gli strumenti necessari e di OpenNebula, reperita direttamente dai repository ufficiali:

```
$ sudo apt-get install g++ libxmlrpc-c3-dev scons libsqlite3-dev libmysqlclient-  
dev libxml2-dev libssl-dev ruby sqlite3 ruby-sqlite3  
$ sudo apt-get install openssh-server  
$ sudo apt-get install ruby1.9.1-full  
$ sudo apt-get install gem  
$ sudo gem install json  
$ sudo mv /usr/lib/ruby/1.9.1/json.rb /usr/lib/ruby/1.9.1/json.rb.no
```

Repository ed OpenNebula:

```
$ wget -q -O- http://downloads.opennebula.org/repo/Ubuntu/repo.key | apt-key add -  
$ echo "deb http://downloads.opennebula.org/repo/Ubuntu/12.04 stable opennebula"  
  > /etc/apt/sources.list.d/opennebula.list  
$ sudo apt-get update  
$ sudo apt-get install opennebula opennebula-sunstone nfs-kernel-server
```

NOTA: I seguenti passaggi sono relativi ad un SO Ubuntu 12.04, per la realizzazione del progetto però è stata utilizzata un'immagine CentOS 6.6 con il software OpenNebula già preinstallato e configurato per un primo utilizzo. Grazie a queste immagini è stato possibile partire già dal passo successivo, con la modifica dei file hosts e network per la costruzione della rete.

3 MODIFICA DEL FILE NETWORK E HOSTS

Proseguendo è necessario effettuare alcune modifiche su due file:

```
/etc/sysconfig/network  
/etc/hosts
```

Nel primo definiamo il nome dell'hostname in cui ci si trova, se siamo nel frontend definiremo quell'host con il nome *frontend*, se siamo nel worker node lo chiameremo *host1*.

Si utilizza vi come editor perchè nano non è supportato sulle macchine CentOS, per sicurezza è conveniente testare sempre di aver fatto le modifiche necessarie tramite un semplice cat:

```
# sudo vi /etc/sysconfig/network
```

Il risultato di questa operazione dovrà essere il seguente (esempio mostrato per il frontend):

```
# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=frontend
```

Nel secondo file andiamo a definire le corrispondenze IP – hostname di tutti gli host coinvolti, ovvero quelli del frontend e quelli dei worker nodes.

Per la scoperta dell'indirizzo ip su una macchina CentOS il comando terminale è il seguente:

```
# ip address show
```

Il comando per la modifica del file hosts è la medesima del file network:

```
# sudo vi /etc/hosts
```

Il risultato di questa operazione dovrebbe avere un output come il seguente con le corrette corrispondenze IP- host:

```
# cat /etc/hosts
127.0.0.1 localhost ...
172.16.209.128 host1
172.16.209.129 frontend
```

Quando le modifiche sono state apportate possiamo testare la risoluzione IP – hostname tramite dei ping, verificando se dal frontend vengono raggiunti i worker nodes e viceversa.

4 TUNNEL SSH

I comandi seguenti mostrano come impostare un tunnel SSH tra due macchine coinvolte nel cloud, esse hanno necessità di comunicare e possono farlo senza una password di autenticazione:

```
# ssh-keygen
# ssh-copy-id root@host1
```

I comandi devono essere effettuati su tutti i nodi coinvolti, se ci si trova su un workernode l'hostname del secondo comando sarà quello del frontend.

Nel primo comando, dopo l'invio, verranno chieste delle informazioni in input, per far sì che l'autenticazione avvenga senza password si deve inserire solo il primo comando dove viene chiesto dove salvare la key, tutte le altre richieste devono essere lasciate vuote.

5 NFS

Il Network File System(NFS), nell'ambito dei sistemi distribuiti, rappresenta un punto di storage e di condivisione tra host. Nella configurazione adottata il datastore è condiviso tra gli host facenti parte del cloud, Questo permette agli host di condividere strutture dati ed in particolare le VMs. I datastore che ci interessa condividere con i worker nodes sono contenuti sul frontend nella directory */var/lib/one/datastore*. Le operazioni necessario per attivare il NFS distribuito i passaggi necessari da effettuare sono i seguenti.

Sul frontend configuriamo il file */etc/exports* come segue:

```
# vi /etc/exports  
  
/var/lib/one *(rw,sync,no_subtree_check)
```

Nel file */etc/fstab* dei worker nodes appendiamo la seguente riga, la quale è necessaria per il raggiungimento della cartella dei datastores condivisa sul frontend:

```
# vi /etc/fstab  
-  
-  
frontend:/var/lib/one    /var/lib/one    nfs    udp,_netdev,nfsvers=3,noauto    0    0
```

Il file è formattato in colonne, quindi tra un parametro e l'altro ricordarsi di mettere un TAB.

Sempre sui worker nodes, creare la cartella che conterrà i datastores:

```
# mkdir -p /var/lib/one
```

Ad ogni avvio e ad ogni operazione significativa apportata potrebbe essere necessario riavviare il server NFS:

```
# /etc/init.d/nfs restart (oppure /sbin/service nfs restart )
```

Infine montiamo la cartella, sui worker nodes:

```
# mount /var/lib/one
```

6 OPENNEBULA SUNSTONE

A questo punto si può controllare se tutte le operazioni sono state effettuate correttamente, accedendo alla SUNSTONE, l'interfaccia grafica di opennebula.

Per accedere ad essa, è necessario riprendere l'indirizzo IP del frontend (se configurato in una macchina virtuale, localhost altrimenti) ed accedervi tramite la porta predefinita 9689; dovrebbe apparire una schermata di login come la seguente :

The image shows the OpenNebula Sunstone login interface. It features the 'OpenNebula Sunstone' logo at the top. Below the logo is a login form with two input fields: 'Username' and 'Password'. There is a 'Keep me logged in' checkbox and a 'Login' button at the bottom of the form.

Username : oneadmin
Password : opennebula

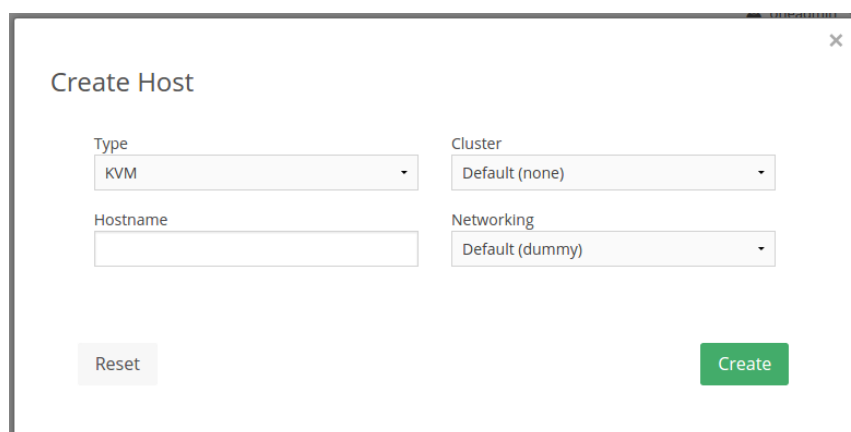
Una volta acceduti le azioni principali da effettuare per completare l'infrastruttura sono principalmente 3 e possono essere tutte fatte tramite interfaccia:

1. Creazione degli Hosts
2. Creazione del Cluster
3. Creazioni delle Vms

6.1 CREAZIONE DEGLI HOSTS

Come primo passaggio andiamo a definire gli hosts coinvolti nell'architettura, ciò può essere fatto tramite la voce "Hosts" sotto "Infrastructure".

Al click sull pulsantino verde dell'interfaccia dovrebbe aprirsi il seguente popup:

The image shows a 'Create Host' popup window. It has a title bar with a close button (X). The form contains four fields: 'Type' (a dropdown menu with 'KVM' selected), 'Cluster' (a dropdown menu with 'Default (none)' selected), 'Hostname' (a text input field), and 'Networking' (a dropdown menu with 'Default (dummy)' selected). At the bottom left is a 'Reset' button, and at the bottom right is a green 'Create' button.

Nel campo *Type* selezioniamo la virtualizzazione KVM, come *Hostname* mettiamo il nome preciso dell'host che si intende creare (quindi avremo un host di nome frontend e uno di nome host1). *Cluster* è un'opzione che verrà modificata successivamente con il nome del cluster che verrà creato al prossimo passo e *Networking* può essere lasciata default.

6.2 CREAZIONE DEL CLUSTER

Ora creiamo un Cluster, il quale contiene gli hosts presenti nel sistema. Questo passo può essere effettuato sotto la voce “Infrastructure” - “Cluster” e nel popup per la creazione basta inserire un nome desiderato ed indicare gli host facenti parte di questo.

6.3 CREAZIONE DELLE VMs

La creazione delle macchine può essere fatta sotto la voce “Virtual Resources” - “Virtual Machines”, alla creazione vengono chiesti un nome e un template (per i nostri scopi può essere utilizzata l'immagine Linux ttylinux preesistente).

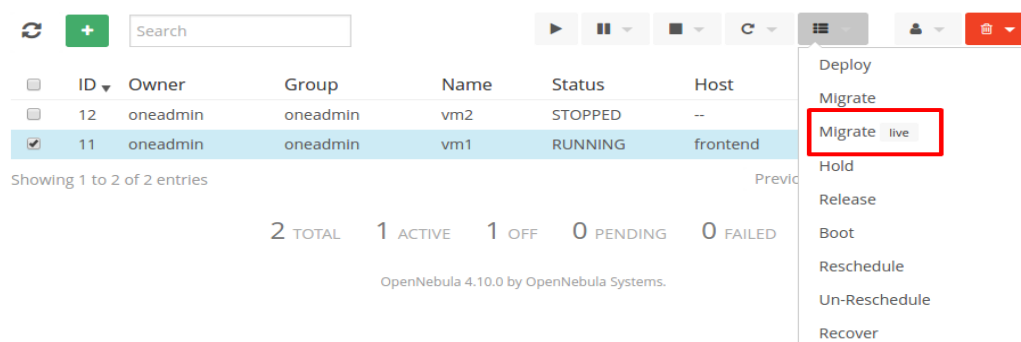
7 LIVE MIGRATION

A questo punto è tutto pronto per l'utilizzo, nella sezione delle macchine virtuali è possibile accenderle, stopparle, spegnerle, accedere alla console tramite VNC e scrivere comandi.


Per verificare la proprietà di live migration è possibile stoppare una VM, così facendo avremo una situazione in cui una VM è in stato RUNNING in un host (ad esempio il frontend) mentre l'altro host (ad esempio host1) è libero da esecuzioni.

In questa situazione possiamo far migrare l'esecuzione dell'unica macchina virtuale in RUNNING dal frontend all'host1.


Anche questa operazione può essere fatta da interfaccia grafica, come mostrato dalla figura sottostante:



Una volta premuto il tasto di Migrate Live si può notare come la macchina si metta in uno stato di Migration per qualche istante:

| | | | | | | | | |
|-------------------------------------|----|----------|----------|-----|---------|-------|----------|---|
| <input checked="" type="checkbox"/> | 11 | oneadmin | oneadmin | vm1 | MIGRATE | host1 | 10.0.1.2 |  |
|-------------------------------------|----|----------|----------|-----|---------|-------|----------|---|

Ed infine andrà in esecuzione definitivamente sull'host1:

| | | | | | | | | |
|-------------------------------------|----|----------|----------|-----|---------|-------|----------|---|
| <input checked="" type="checkbox"/> | 11 | oneadmin | oneadmin | vm1 | RUNNING | host1 | 10.0.1.2 |  |
|-------------------------------------|----|----------|----------|-----|---------|-------|----------|---|

Per verificare che l'esecuzione della VMs non si sia fermata è possibile dare un comando prima della fase di migrazione e verificare che esso sia ancora in funzione. Uno di questi potrebbe essere:

```
# watch -n 1 date
```

8 CONCLUSIONI

Nella parte iniziale del procedimento (capitolo 2) si possono riscontrare numerosi problemi, dovuti soprattutto a mancate dipendenze tra pacchetti.

Questo problema è stato scavalcato utilizzando le immagini Sandbox fornite da OpenNebula già preconfigurate all'utilizzo (ciò ha permesso di iniziare direttamente con i passaggi del capitolo 3).

Per far girare le immagini CentOS 6.6 di frontend e worker node si è utilizzato VMWare 7.1.0, sono state provate versioni precedenti ma si riscontravano problemi.

L'infrastruttura è stata creata e la proprietà di migrazione live appare funzionare correttamente.

Il layout di tastiera delle VMs ttylinux non consentono di individuare alcuni caratteri fondamentale che servirebbero per l'installazione del software JBoss.

9 BIBLIOGRAFIA

1. Guida ufficiale per installazione di OpenNebula 4.4 su Ubuntu 12.04 :
http://docs.opennebula.org/4.4/design_and_installation/quick_starts/qs_ubuntu_kvm.html
2. Guida alla modifica di alcune impostazioni network su CentOS:
http://www.rackspace.com/knowledge_center/article/centos-hostname-change
3. Guida all'utilizzo delle Sandbox OpenNebula:
<http://opennebula.org/tryout/sandbox-testdrive/>