

Segesdi Dániel

OpenNebula

Virtualizációs technológiák és alkalmazásaik

BMEVIMIAV89

2011 ősz

OpenNebula

Előszó

A feladatom az OpenNebula nyílt forráskódú cloud management eszköz megismerése, mely egységes felületen keresztül teszi elérhetővé a különböző cloudok szolgáltatásait. Sajnos a kipróbálására nem volt lehetőségem, mivel nem áll rendelkezésemre megfelelő hardver. A továbbiakban az alapfogalmakról, a szükséges hardveres és szoftveres háttérrel, és az elérhető funkciókról lesz szó.

Általában a cloudokról

A cloudok alapvetően három csoportba sorolhatóak aszerint, hogy milyen szolgáltatást nyújtanak a kliensek számára: SaaS, PaaS, IaaS

SaaS=Software as a Service: Különböző, a szerverekre telepített alkalmazásokhoz nyújt hozzáférést. Így könnyebb a szoftver frissítéseket elvégezni, jobban kihasználhatóak az erőforrások.

PaaS=Platform as a Service: Egy komplett platformhoz nyújt hozzáférést, ami magában foglal valamilyen számítógépes architektúrát, fájlrendszert, operációs rendszert, felhasználói felületet.

IaaS=Infrastructure as a Service: Egy számítógépes infrastruktúrát szolgáltat, nyers tárhelyet (blokkok) és valamilyen hálózatot. Tipikusan virtuális gépek/platformok futtatására használják.

A cloudokat hozzáférési modellek alapján is csoportosíthatjuk:

Private: Egy szervezet számára fenntartott számítási felhő. Elsősorban a cloudok rugalmasságát használja ki, szolgáltatásokat és egyéb szoftvereket tesznek elérhetővé.

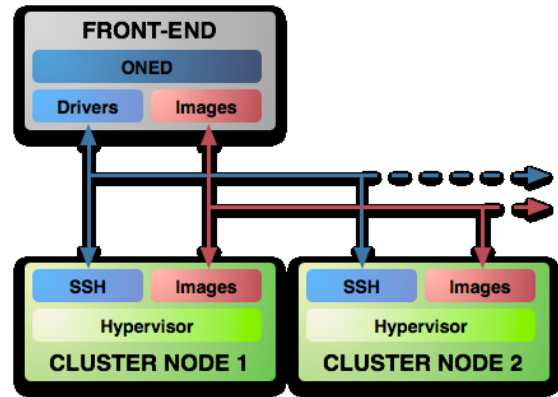
Public: Ezek a hagyományos, legelterjedtebb cloudok. Dinamikus oszt szét erőforrásokat az interneten bárkinek webes szolgáltatásokon és alkalmazásokon keresztül. A felhasznált erőforrások alapján történhet a számlázás.

Hybrid: Összetett számítási felhő, ami több privát és nyilvános cloud összessége. Az egyes cloudok megtartják különálló létüket, de összekapcsolódnak egymással, így mind a privát, mind a publikus felhők előnyeit nyújtva. Úgy is definiálható, mint egy több cloudból álló rendszer, amik között könnyen mozgathatók adatok illetve alkalmazások.

Az OpenNebula privát, nyilvános és hibrid, IaaS típusú felhők létrehozására alkalmas teljesen nyílt forráskódú eszköztár.

A rendszer felépítése

Az OpenNebula működéséhez egy tényleges számítógépekből álló clusterre (*fürtre*) van szükség. Ez áll egy úgynevezett front-end gépből, hostokból, egy image repositoryból, és a hálózattól. A front-enden működnek az OpenNebula szolgáltatásai, a hypervisorokat futtató hostok pedig a számítási felhő virtuális gépek által igényelt erőforrásait adják. Az image repository bármilyen tárolóeszköz lehet, melyen a virtuális gépek alapvető image fájlljai találhatóak meg, a hálózat pedig egy tényleges fizikai hálózatot jelent, mely rendelkezik VLAN támogatással, tehát képes virtuális gépek között kapcsolatot létesíteni, illetve egyéb szolgáltatások is elérhetőek rajta keresztül (pl.: tárhely).



Front-End

A számítógép, melyre az OpenNebulát telepítjük a front-end. Ennek a gépnek hozzáféréssel kell rendelkeznie az image repositoryhoz (helyi, vagy hálózati is lehet), és hálózati kapcsolattal minden hosthoz. Az OpenNebula alap változata kevesebb, mint 10 MB helyet foglal telepítés után.

Az OpenNebula szolgáltatásai közé tartoznak:

- Menedzsment rendszer, és ütemező
- Monitorozó és számlázó rendszer
- Webes interfész szerver
- Cloud API szerverek

Hostok

A hostok fizikai gépek, amiken a virtuális gépek futnak. A rendszer telepítése során megfelelően konfigurálni kell az OpenNebulát, hogy képes legyen SSH kapcsolat létrehozására a hostokkal, és engedélyezni kell, hogy futtathasson parancsokat a hostokon adminisztrátori jogokkal. Az OpenNebula semmilyen komponens telepítését nem igényli a hostokra, csak az alábbi megkötések érvényesek:

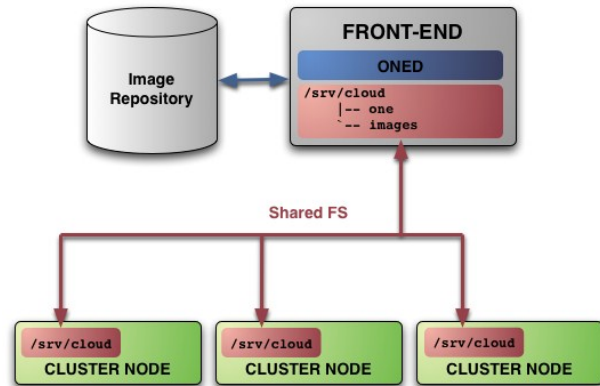
- SSH szerver futtatása
- megfelelően konfigurált, támogatott hypervisor

Érdekes lehet, hogy mivel az OpenNebula több féle hypervisort támogat, lehetőség van heterogén clusterek létrehozására ezekből. Sajnos arra vonatkozóan nem találtam információt, hogy milyen funkciók működnek különböző hypervisorok között, valószínűleg az azonos típusokat egy egységként kezeli a rendszer, és a terheléselosztás, migration, stb. csak ilyen egységeken belül működik.

Image repository és tárhely

Az OpenNebula egy gyűjteményt használ a virtuális gépek image fájljainak tárolására. Ezt a front-enden keresztül elérhetővé kell tenni bármilyen megfelelő technológia alkalmazásával (NAS, SAN, vagy közvetlenül csatolt tárhely)

Az image fájlkat el kell juttatni a hostokhoz, hogy használni tudják azokat. Az OpenNebula több féle szervezést is kezelni tud. Ezek közül az egyik egy megosztott fájlrendszer a front-end és a hostok között. Ilyen módon a hypervisorok minden szolgáltatását kihasználhatjuk (pl.: live migration), illetve általában jobb VM indítási sebességet eredményez.



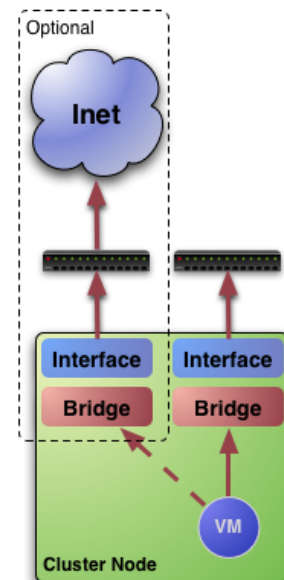
Általában a tárolt image fájlkat másolni(clone) kell, mielőtt a hostok használhatnák, ezért az image repository méretét annak megfelelően kell meghatározni, hogy mennyi virtuális gépet fogunk futtatni a clusteren.

Hálózat

Az OpenNebula biztosít számunkra egy testreszabható hálózati alrendszert. A hálózatra a hostok kezelése, a hypervisorok monitorozása és az image fájl mozgatása miatt van szükség. Infrastruktúrától függően előnyös lehet egy dedikált hálózat létrehozása erre a célra.

Hogy a különböző hostokon futó VM-ek között kapcsolatot tudjunk létesíteni, a virtuális gép hálózati interfészét egy a fizikai gépben lévő ethernet hídhoz kell csatlakoztatni. Hogy hatékonyan kihasználható legyenek a virtuális gépeink, lehet, hogy több fizikai hálózathoz is csatlakoztatnunk kell őket.

Például egy tipikus host két fizikai hálózattal: az egyik a publikus IP hálózathoz, a másik pedig a virtuális LAN-ok részére.



Egy privát felhő

A következő részben egy rövid példán keresztül bemutatom, hogyan lehet egy egyszerű számítási felhőt használni, azon különböző parancsokat végrehajtani az OpenNebula segítségével.

Privát felhőket általában vállalaton belül szoktak használni, mert hatékonyan és rugalmasan kielégíthetők a felhasználók folyamatosan változó erőforrásigényei. Különböző szolgáltatások futnak virtuális gépeken, melyeket az OpenNebula különböző felületein át lehet megfigyelni és menedzselni.

Egy példa a cloud használatára a CLI-n keresztül (Command Line Interface):

Először is kérdezzük le, hogy milyen hostok találhatóak a clusterben! Ezt az *onehost list* paranccsal tehetjük meg:

```
$ onehost list
ID NAME          CLUSTER    RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
0  host01        default    0    800   800   800  8194468 7867604 on
1  host02        default    0    800   797   800  8387584 1438720 on
```

Láthatjuk, hogy a példában 2 hosttal rendelkezünk.

Ezek után regisztrálhatunk egy image fájlt az *oneimage register* paranccsal. Hogy ezt megtehessek, létrehozunk egy image template-t, az előzőleg a */home/cloud/images* mappában elhelyezett fájlhoz. Az *ubuntu.oneimg* template tartalma:

```
NAME          = "Ubuntu Desktop"
PATH          = /home/cloud/images/ubuntu-desktop/disk.0
PUBLIC        = YES
DESCRIPTION   = "Ubuntu 10.04 desktop for students."
```

Ezek után végezzük el a regisztrációt, majd utána kérdezzük le az összes beregisztrált image-t:

```
$ oneimage register ubuntu.oneimg
$ oneimage list
ID  USER      NAME TYPE      REGTIME  PUB  STAT  #VMS
1  oneadmin  Ubuntu Desktop  OS    Jul 11, 2010 15:17  Yes  rdy   0
```

Ahhoz, hogy ezt használni tudjuk egy virtuális gépben, létrehozunk egy VM template-t:

```
CPU          = 1
MEMORY       = 2056

DISK = [ image = "Ubuntu Desktop" ]

DISK = [ type   = swap,
         size   = 1024 ]

NIC          = [ NETWORK = "Public network" ]
```

Itt beállítjuk, hogy milyen erőforrással kell rendelkeznie a virtuális gépünknek, illetve hogy melyik image-t használja.

Ha beállítottuk a kívánt értékeket (elsősorban CPU és memória), és ellenőriztük, hogy legalább az egyik host tudja így futtatni a Vm-et, továbbléphetünk. Az `onevm create` paranccsal példányosítjuk a VM-et:

```
$ onevm create myFirstVM.template
```

Ez után a Vm kap egy ID-t, amivel később hivatkozhatunk rá. A létrehozott VM-ek listája:

```
$ onevm list
ID      USER      NAME  STAT CPU    MEM      HOSTNAME  TIME
0  oneadmin  one-0  runn  0    65536   host01   00 0:00:02
```

A listából megtudjuk a VM ID-jét, állapotát (STAT) - jelenleg `runn`, azaz futó állapotban van - hogy mennyi ideje fut a VM és hogy épp melyik hoston fut. Ezek után, ha ismerjük a VM IP-címét, be is jelentkezhetünk.

Egy másik hasznos funkció a migration, amivel azonos hypervisorral rendelkező hostok között mozgathatunk VM-eket. Ennek egy speciális esete a live migration, amikor ezt a VM leállítása nélkül, folyamatos működés mellett hajtjuk végre:

```
$ onevm livemigrate 0 1
```

Az `onevm livemigrate` parancs az áthelyezendő VM, és a cél host ID-jét várja paraméterül. Láttuk, hogy a VM ID-je 0, és a host01 nevű hoston fut. Ezt helyezzük át most az 1-es ID-jű host02 nevű számítógépre.

Sunset GUI

Ugyanezeket a funkciókat természetesen nem csak parancssorból érhetjük el, hanem például a sokkal kényelmesebb internetes felhasználói felületen keresztül is.

The screenshot shows the OpenNebula Sunstone web interface. The top navigation bar includes 'OpenNebula Sunstone', 'Documentation | Support | Community', and 'Welcome oneadmin | Sign Out'. The left sidebar has a menu with 'Dashboard', 'Hosts & Clusters' (highlighted), 'Virtual Machines', 'Virtual Networks', 'Images', and 'Users'. The main content area displays a table of VMs with columns for 'Show', 'ID', 'Name', 'Cluster', 'Running VMs', 'CPU Use', 'Memory use', and 'Status'. Below the table, there are tabs for 'Host information' and 'Host template'. The 'Host information' tab is active, showing details for host 'hostE' (ID: 37, State: MONITORED, Cluster: default, IM MAD: im_dummy, VM MAD: vmm_dummy, TM MAD: tm_dummy). To the right, 'Host shares' are listed: Max Mem: 16G, Used Mem (real): 0K, Used Mem (allocated): 0K, Used CPU (real): 0, Used CPU(allocated): 0, and Running VMs: 0. The footer contains the copyright notice: 'Copyright 2002-2011 © OpenNebula Project Leads (OpenNebula.org). All Rights Reserved. OpenNebula 2.3.0'.

Értékelés

Egy nagyon jól megvalósított rendszernek tűnik, óriási rugalmassággal. Elégé elterjedt az utóbbi pár évben, valószínűleg annak köszönhetően, hogy nagyon sok szolgáltatást, interfészt, API-t, felhasználói felületet támogat, és ha mégis más igényünk van, akkor is írhatunk hozzá saját plugineket. A kezelése egyszerűnek tűnik, bár kipróbálni nem volt lehetőségem, így csak a leírások alapján tudok erről nyilatkozni. Nagyon sajnálom, hogy nem állt rendelkezésemre megfelelő hardveres környezet, de így is sok mindent megtudtam a rendszerről, illetve a témáról.

Felhasznált források

- <http://opennebula.org> – hivatalos honlap, dokumentáció és útmutatók
- <http://en.wikipedia.org>