

OpenStack/Opennebula: Infraestructures com a servei (IaaS)

Autor:

Jordi Claret Membrado
ETIS

Consultor:

Joaquin Lopez Sanchez-Montañes

Sabadell, 13 Juny 2011

Índex de continguts

1.-Dedicatòria i agraïments.....	4
2.-Propòsit.....	4
3.-Objectius del TFC.....	4
4.-Enfocament i mètode seguit.....	4
5.-Planificació del projecte.....	5
6.-Breu descripció dels altres capítols de la memòria.....	6
7.-Cloud Computing – Definicions.....	7
7.1.- Característiques essencials.....	7
7.2.- Models de servei.....	8
7.3.- Models de desplegament Cloud.....	10
8.-Cloud Computing – Evolució, no revolució (Evolution, not Revolution).....	10
9.-IaaS. Infraestructura com a servei.....	13
9.1.- IaaS – Productes.....	13
9.1.1.- OpenNebula.....	13
9.1.1.1.-Introducció.....	13
9.1.1.2.-Característiques.....	14
9.1.1.3.-Arquitectura.....	15
9.1.1.3.1.-Interna.....	15
9.1.1.3.2.-Infraestructura bàsica.....	17
9.1.1.4.-Processos i directoris.....	18
9.1.1.5.-Xarxes.....	20
9.1.1.1.1.-Storage.....	21
9.1.1.2.-Cicles de vida.....	22
9.1.1.2.1.-Imatges virtuals.....	22
9.1.1.2.2.-Màquines virtuals.....	22
9.1.1.3.-Fitxers.....	24
9.1.1.3.1.-Configuració.....	24
9.1.1.1.1.1.-Logs.....	25
9.1.1.2.-Instal·lació Front-End i WorkNodes.....	25
9.1.1.1.1.-Sunstone: Cloud Operations Center.....	30
9.1.2.-OpenStack.....	33
9.1.2.1.-Característiques.....	33
9.1.2.2.-Arquitectura.....	33
9.1.2.3.-OpenStack Compute.....	34
9.1.2.3.1.1.-Hipervisors.....	34
9.1.2.3.1.2.-Usuaris i projectes.....	35
9.1.2.3.1.3.-Imatges i instàncies.....	36
9.1.2.3.1.4.-Arquitectura interna.....	37
9.1.2.3.1.5.-Storage.....	39
9.1.2.3.1.6.- Instal·lació OpenStack Compute.....	39
9.1.2.3.1.7.-Fitxers configuració.....	44
9.1.2.4.-OpenStack Object Storage.....	45
9.1.2.4.1.-Objectes i contenidors.....	45
9.1.2.4.2.-Instal·lació Swift.....	45

9.1.2.4.3.- <u>OpenStack Dashboard</u>	51
9.1.3.- <u>Taula Comparativa</u>	53
9.1.4.- <u>Conclusions</u>	54
9.1.4.1.- <u>Disseny tècnic. Instal·lació Servidors Linux Ubuntu</u>	56
10.- <u>Bibliografia</u>	58

1.- Dedicatòria i agraïments

La meva dedicatòria i agraïments a les comunitats open-source que contribueixen al desenvolupament i al creixement de les societats. També especial menció als creadors de llocs wikis i blogs que tant es ajuden quan busquem una solució a un problema o senzillament contribuint a que tots tinguem accés a la cultura.

2.- Propòsit

El propòsit d'aquest document és endinsar-nos en els nous models de negoci basats en les tecnologies Cloud Computing amb l'objectiu de consolidar nous coneixements complementaris als estudis d'enginyeria informàtica. Es vol realitzar un estudi de productes open-source basats en un model de servei com a infraestructura (IaaS), i, implementar-los des de la vessant més pràctica.

3.- Objectius del TFC

L'objectiu principal d'aquest Treball Final de Carrera és el d'assimilar coneixements sobre Cloud Computing posant al microscopi alguns dels productes open-source amb més rellevància i estudiar la seva arquitectura i filosofia.

4.- Enfocament i mètode seguit.

L'enfocament donat al treball és principalment pràctic, però entenent el conceptes teòrics, passant per la configuració hardware, instal·lació d'un sistema operatiu Linux i la configuració dels diferents serveis per posar en marxa un cloud privat. La recerca per internet i l'estudi de la documentació

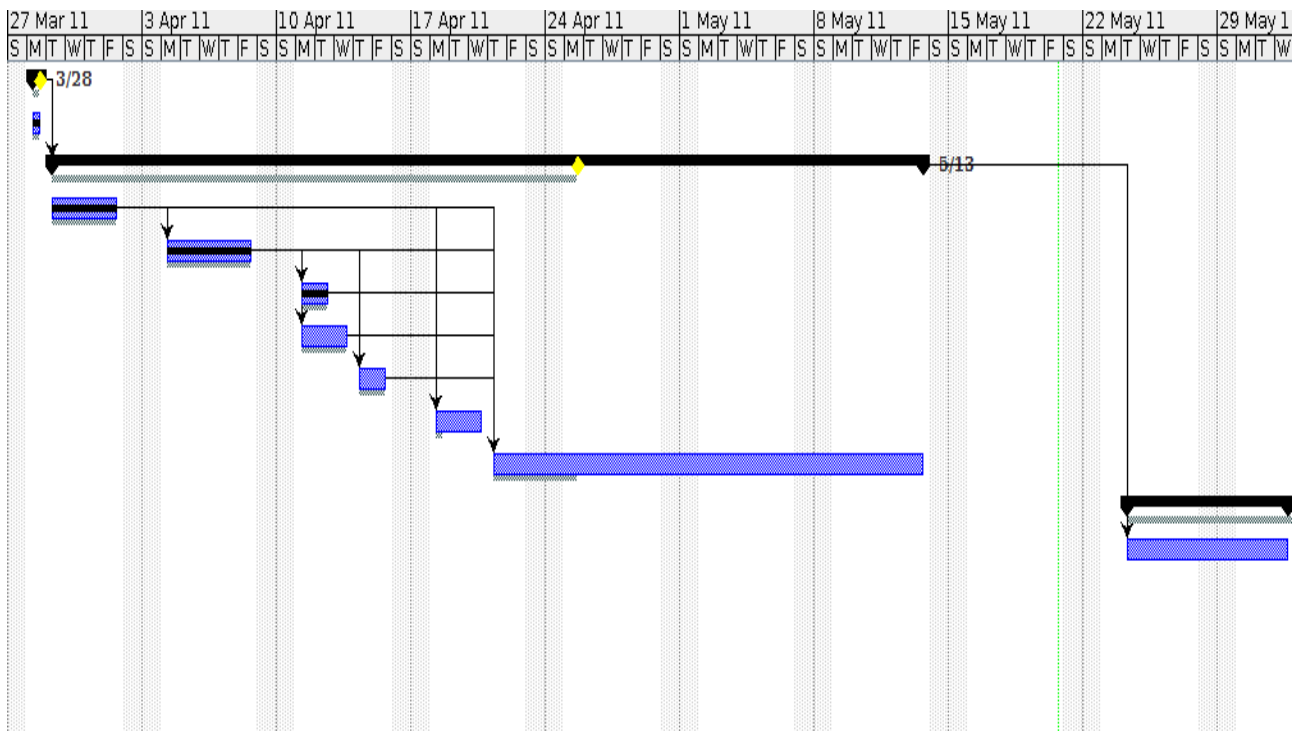
trobada als portals dedicats a clouds ha estat una part fonamental, tot i que de vegades, aquesta feina no es vegi reflexada de forma escrita.

El mètode seguit ha estat basat en l'estudi de la documentació que trobem a les webs oficials dels productes i la pràctica dels mateixos, complementant la documentació tècnica amb la consulta de llistes de distribució o fòrums dedicats a temàtiques cloud.

5.- Planificació del projecte.

El seguiment i la planificació del projecte han estat elaborats amb l'eina OpenProj, seguint una sèrie de punts estratègics o fites marcades per a la consecució final del treball. Tot seguit es mostra la planificació del treball en forma de captura d'imatge directament obtinguda de OpenProj.

	☺	Name	Duration	Start	Finish
1	✓	☐ PAC1	1 day	3/28/11 8:00 AM	3/28/11 5:00 PM
2	✓	Descripció i planificacio del projecte	1 day	3/28/11 8:00 AM	3/28/11 5:00 PM
3	☐	☐ PAC2 i PAC3	20 days	3/29/11 8:00 AM	4/25/11 5:00 PM
4	✓	Estudi comparativa paquets Private Cloud Linux	4 days	3/29/11 8:00 AM	4/1/11 5:00 PM
5	☐ ✓	Instal.lacio Private Cloud	5 days	4/4/11 8:00 AM	4/8/11 5:00 PM
6	✓	Configuració hipervisors	2 days	4/11/11 8:00 AM	4/12/11 5:00 PM
7	☐	Configuració xarxa	3 days	4/11/11 8:00 AM	4/13/11 5:00 PM
8	☐	Configuració emmagazament	2 days	4/14/11 8:00 AM	4/15/11 5:00 PM
9	☐	Informe grau progrés. Desviacions i mesures.	1 day	4/18/11 8:00 AM	4/18/11 5:00 PM
10	☐	Elaboració Memòria	3 days	4/21/11 8:00 AM	4/25/11 5:00 PM
11	☐	☐ Entrega Final	7 days	6/3/11 8:00 AM	6/13/11 5:00 PM
12	☐	Elaboració Memòria	7 days	6/3/11 8:00 AM	6/13/11 5:00 PM
13	☐	☐ Presentacio i video	4 days	6/14/11 8:00 AM	6/17/11 5:00 PM
14		Elaboració Presentacio i video	4 days	6/14/11 8:00 AM	6/17/11 5:00 PM



6.- Descripció dels altres capítols de la memòria.

Els primers capítols són introductoris explicat l'enfocament, mètode i planificació seguits durant el projecte. Els següents capítols mostren les característiques essencials dels conceptes de cloud computing i d'una infraestructura com a servei (IaaS). Els capítols principals mostren una descripció conceptual i una instal·lació en entorns de testing de dos productes IaaS com són: opennebula i openstack. Els capítols finals mostren una taula comparativa i les conclusions.

7.- Cloud Computing – Definicions.

Cloud computing és un model per habilitar de forma convenient, l'accés a xarxes sota demanda a un pool compartit de recursos configurables, com ara xarxes, servidors, emmagatzematge, aplicacions i serveis, que poden ser ràpidament aprovisionats o alliberats amb un esforç mínim de gestió.

Aquest model està format per :

- 4 característiques essencials
- 3 models de servei
- 4 models de desplegament.

7.1.- Característiques essencials

El Cloud computing ens crea la il·lusió d'infinits recursos disponibles sota demanada, eliminant la necessitat d'usuaris cloud computing per a programar l'aprovisionament. Les principals característiques són:

- **On-demand Self-Service.** Els usuaris poden unilateralment aprovisionar recursos de computació, com servidors o emmagatzematge en xarxa, sense la necessitat de l'acció de cap tècnic de sistemes.
- **Pool de recursos.** Els recursos de computació d'un proveïdor estan agrupats per a servir a múltiples consumidors/clients fent ús d'un model de pagament per servei amb diferents recursos virtuals i/o físics dinàmicament assignats i/o re-assignats/alliberats sota demanda per cada client. Tot i que els usuaris poden tenir la sensació de que en cap moment saben on estan ubicats geogràficament els recursos dels que fan ús, poden especificar amb un alt nivell d'abstracció (ex. país, data-center, etc..). Exemples de recursos són emmagatzematge, processadors, memòria, xarxa, ample de banda i màquines virtuals.

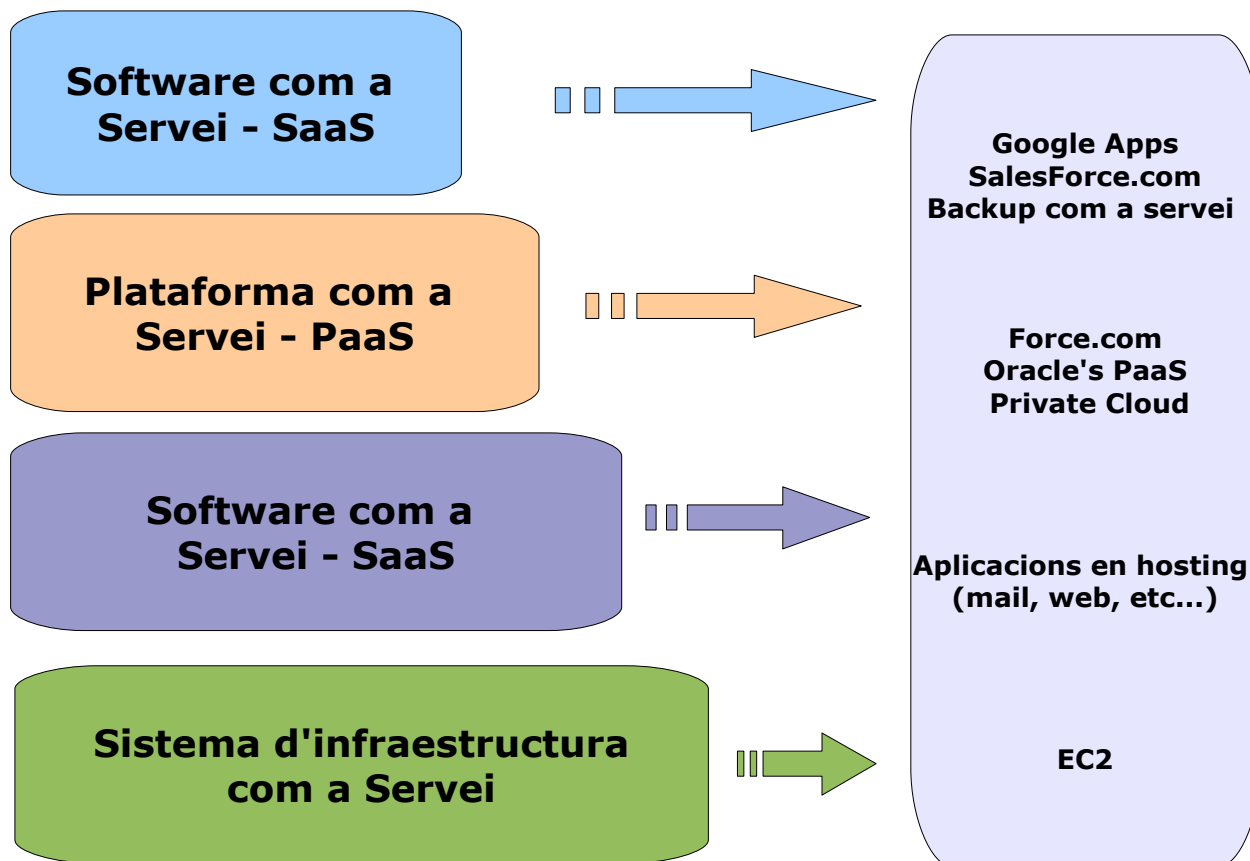
- **Escalabilitat.** Les capacitats necessàries perquè els usuaris puguin ràpidament i elàsticament aprovisionar recursos, i en alguns casos, de forma automàtica, per a escalar ràpidament amunt o avall. Per als consumidors, les capacitats disponibles per aprovisionar són il·limitades i poden comprar recursos en qualsevol quantitat i en qualsevol moment.
- **Servei moderat.** En sistemes que formen el Cloud computing controlen i optimitzen els recursos de forma automàtica. Aquests recursos poden ser monitoritzats, controlats, i reportats per analitzar el seu ús per part del proveïdor i/o consumidors/clients.

7.2.- Models de servei

- **Infraestructura Cloud com a servei – IaaS.** Les capacitats proveïdes al consumidors són capacitats de processament, emmagatzematge, xarxes, i d'altres recursos que inclouen sistemes operatius i aplicacions. Els consumidors no tenen cap control ni gestionen la infraestructura cloud, però si ténen control sobre els sistemes operatius, emmagatzematge, aplicacions desplegades i possibilitat d'un control limitat sobre alguns recursos de xarxa.
- **Plataforma Cloud com a servei – PaaS.** Les capacitats proveïdes al consumidors són les de desplegar aplicacions a la infraestructura cloud creada fent ús de llenguatges de programació i diferents eines suportades pel proveïdor. Els consumidors no tenen cap control ni gestionen la infraestructura cloud, però tenen control sobre les aplicacions i sobre l'entorn de configuració de les aplicacions desplegades.
- **Software Cloud com a servei – SaaS.** Les capacitats proveïdes al consumidors són l'ús de les aplicacions que funcionen sobre la infraestructura cloud. Les aplicacions són accessibles des de diversos

dispositius funcionant com a client a través d'una interfície com un navegador (ex webmail). Els consumidors no tenen cap control ni gestionen la infraestructura cloud i només tenen accés a la aplicació i a les configuracions pròpies d'aquestes.

- El següent gràfic mostra exemples de serveis Cloud computing



7.3.- Models de desplegament Cloud.

Els models per desplegar una plataforma cloud són els següents :

- **Cloud Privat.** La infraestructura Cloud opera només per una organització que gestiona la mateixa, o la administren tercers, i està ubicat al centre o centres de dades de la organització.
- **Cloud Públic.** La infraestructura cloud està a disposició del públic i és propietària una organització que ven serveis cloud.
- **Cloud Híbrid.** La infraestructura cloud està composta per dos o més infraestructures cloud (privada, comunitat, o pública) per habilitar la portabilitat de les aplicacions (load-balancing entre clouds).
- **Cloud Comunitari.** La infraestructura cloud és compartida per algunes organitzacions i dona suport a alguna comunitat que comparteix projecte o idea.

8.- Desplegament – iEvolució, no revolució !

El cloud computing requereix de lles diferents tècniques de virtualització com a tecnologia essencial, per aportar unes capes d'abstracció que gestionaran la complexitat i l'escalabilitat de la infraestructura.

Els diferents nivells d'abstracció que trobem són:

- Self-Service
- Serveis elàstics

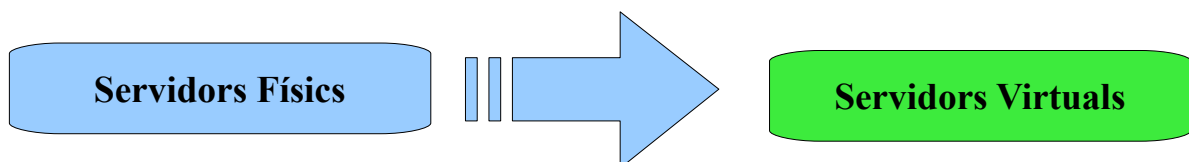
- Independència de la localització d'emmagatzematge i serveis
- Usuaris i grups
- Auditories
- API's, drivers, eines
- Federació

I les relacionem directament amb:

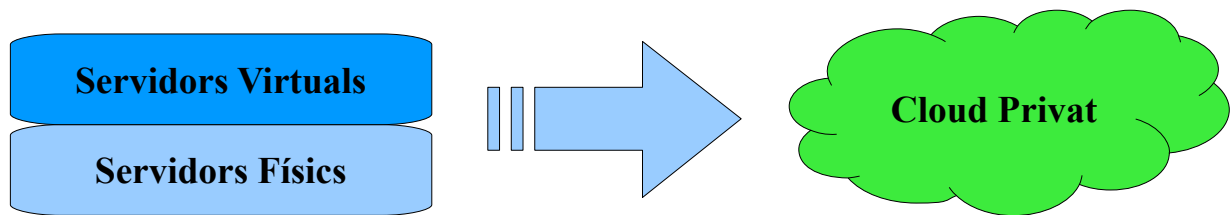
- Bare-Metal o Maquinari : Accés total i sense nivell d'abstracció.
- Virtualització : Abstraccio Hardware o maquinari. Primer nivell d'abstracció que mapeja els recursos físics a virtuals.
- Cloud: Abstracció de recursos. Segon nivell d'abstracció que mapeja el cloud amb recursos virtuals.

Per aconseguir l'objectiu, que no és més, que la unió d'aquests conceptes en forma de cloud com a resultat final, cal seguir les següents fases com a base:

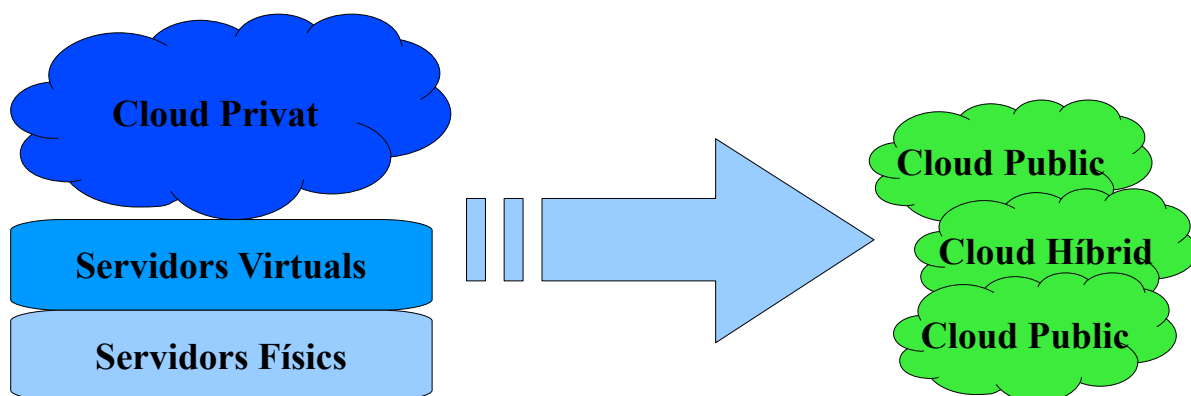
- **Fase 1 : Consolidació de servidors usant la virtualització.**



- **Fase 2 : Automatitzar grans desplegaments dins un cloud privat.**



- **Fase 3 : Extendre el cloud privat fins a clouds híbrids o públics.**



9.- IaaS. Infraestructura com a servei.

Tal i com hem vist en punts anteriors, el model IaaS és l'escollit per l'estudi a detall.

9.1.- IaaS – Productes.

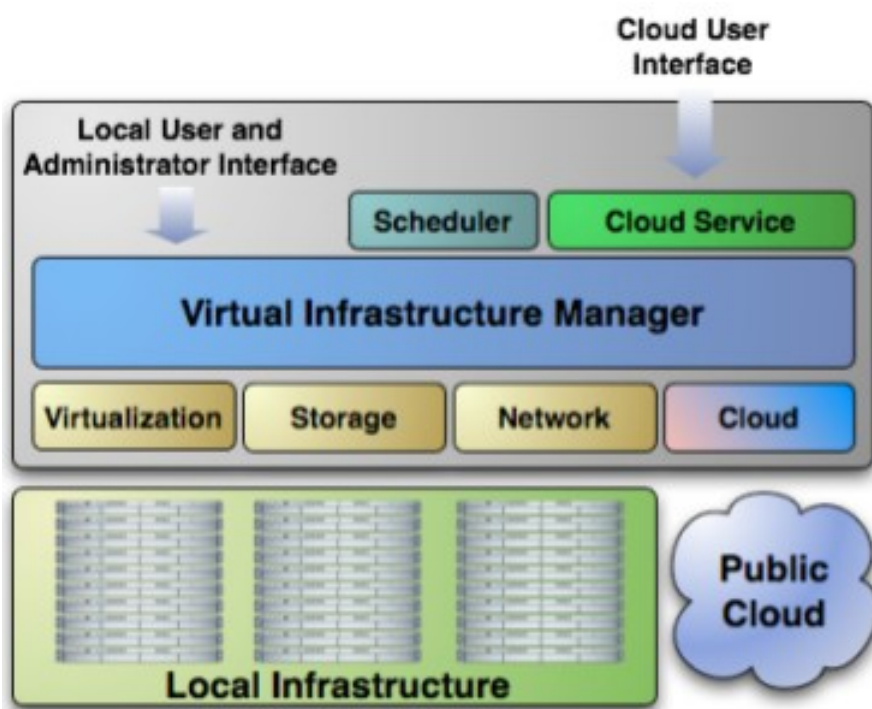
Els programaris amb els que aplicarem la teoria són específics per a la construcció de diferents tipus de clouds :

- OpenNebula (<http://www.opennebula.org/>)
- OpenStack (<http://www.openstack.org/>)

9.1.1.- OpenNebula

9.1.1.1.- Introducció

OpenNebula és un conjunt d'eines open-source per a construir Clouds IaaS privats, públics i híbrids. El primer pas és sempre la construcció de d'un Cloud privat que conseqüentment pot ser extés com a híbrid combinant recursos locals del Cloud privat amb recursos de proveïdors de Cloud remots o com a Cloud públic exposant les interfícies del Cloud per ser accedits per altres Clouds privats o híbrids. OpenNebula gestion el emmagatzematge, les xarxes, tecnologies de virtualització.



Un Cloud privat amb OpenNebula proporciona una infraestructura d'usuaris amb una plataforma elàstica per a un ràpid desplegament i escalabilitat de serveis per atendre a peticions dinàmiques de serveis per part d'usuaris finals. Els serveis funcionen a servidors virtuals (VM - Virtual

Machine) monitoritzats i controlats al Cloud pel front-end OpenNebula Operations Center o qualsevol interfície dels que disposa Opennebula :

- Línia de comandes o Command Line Interface (CLI)
- XML-RPC API
- OpenNebula Cloud APIs
- Libvirt virtualització API

9.1.1.2.- Característiques

Que podem fer amb Opennebula:

- Gestió de la xarxa, servidors, i emmagatzematge.
- Gestió del cicle de vida dels servidors virtuals
- Gestió de la càrrega operacional
- Gestió de xarxes virtuals
- Gestió de imatges
- Gestió de la informació i auditoria
- Gestió de la seguretat
- Gestió de la capacitat de clouds remots
- Gestió de la capacitat de clouds públics

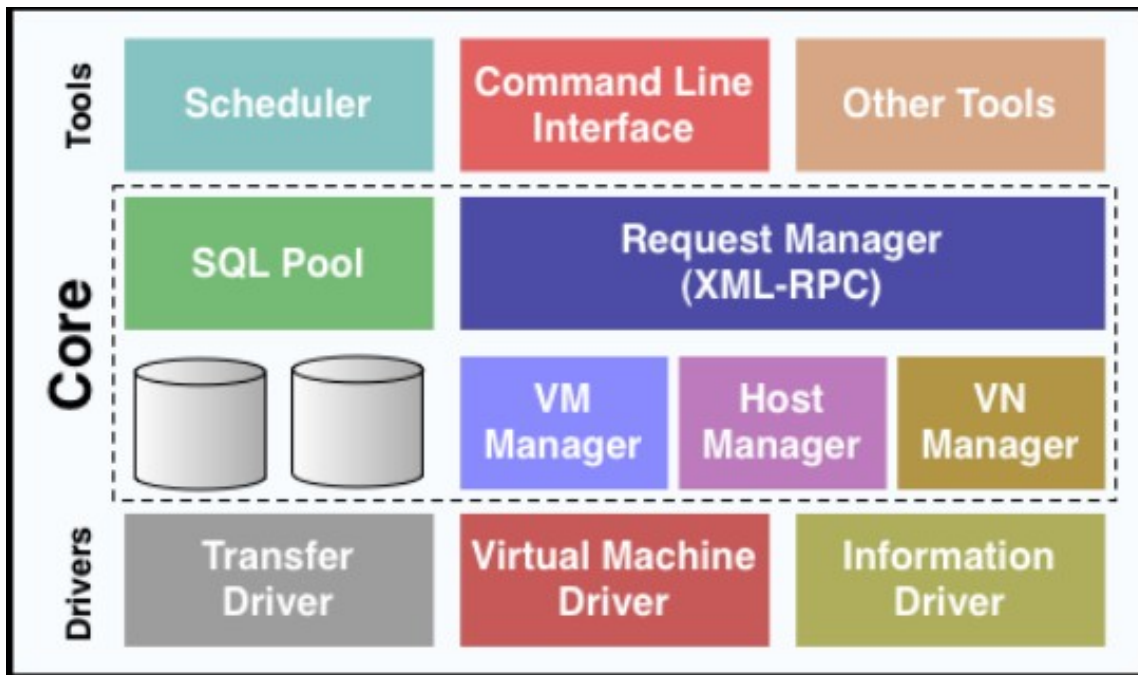
9.1.1.3.- Arquitectura

9.1.1.3.1.- Interna

La arquitectura interna d'OpenNebula és divideix en 3 parts :

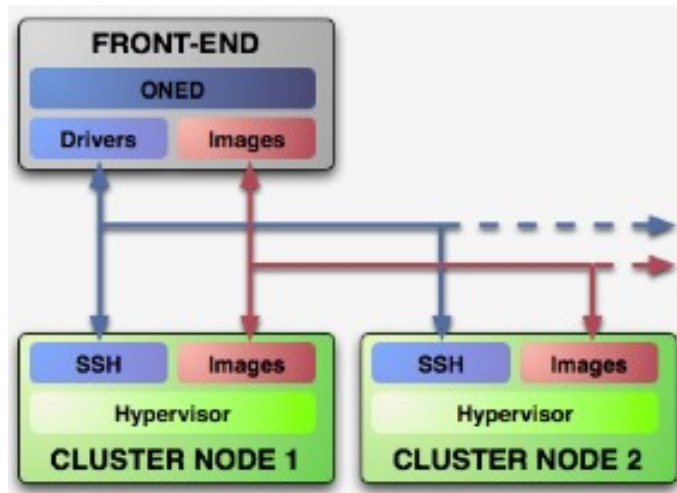
- **Tools.** Eines de gestió desenvolupades fent ús de les interfícies proporcionades pel nucli. Les eines que trobem són CLI (línia de comandes), el planificador, API libvirt, així com d'altres eines de tercers que fan ús de la interfície XML-RPC.
 - CLI. Permet als usuaris i administradors gestionar des de la línia de comandes tota la infraestructura virtual.
 - Planificador. És un component independent a la arquitectura de Opennebula que fa ús de XML-RPC per invocar les accions que han de dur a terme les màquines virtuals.
- **Core.** Components principals per a gestionar màquines virtuals, xarxes virtuals, emmagatzematge i nodes. Els components principals són:
 - Request Manager, gestiona les peticions dels usuaris
 - Virtual Machine Manager, gestiona i monitoritza les màquines virtuals
 - Transfer Manager, gestiona les transferències de màquines
 - Virtual Network Manager, gestiona les xarxes virtuals.
 - Host Manager, gestiona i monitoritza els recursos físics

- Database, base de dades persistent que emmagatzema les dades de opennebula (sqlite o mysql)
- **Drivers.** Proporcionen noves tecnologies per a la virtualització, el emmagatzematge, la monitorització i els serveis de cloud. Interactua amb hipervisor, serveis del cloud, transferència d'imatges o serveis d'informació.



9.1.1.3.2.- Infraestructura bàsica

La instal·lació més bàsica de OpenNebula assumeix que la infraestructura bàsica d'un cluster amb un front-end i almenys, un parell de nodes del cluster on funcionaran el Hosts que gestionaran els Guests o servidors virtuals. La següent imatge mostra el components bàsics de la infraestructura OpenNebula



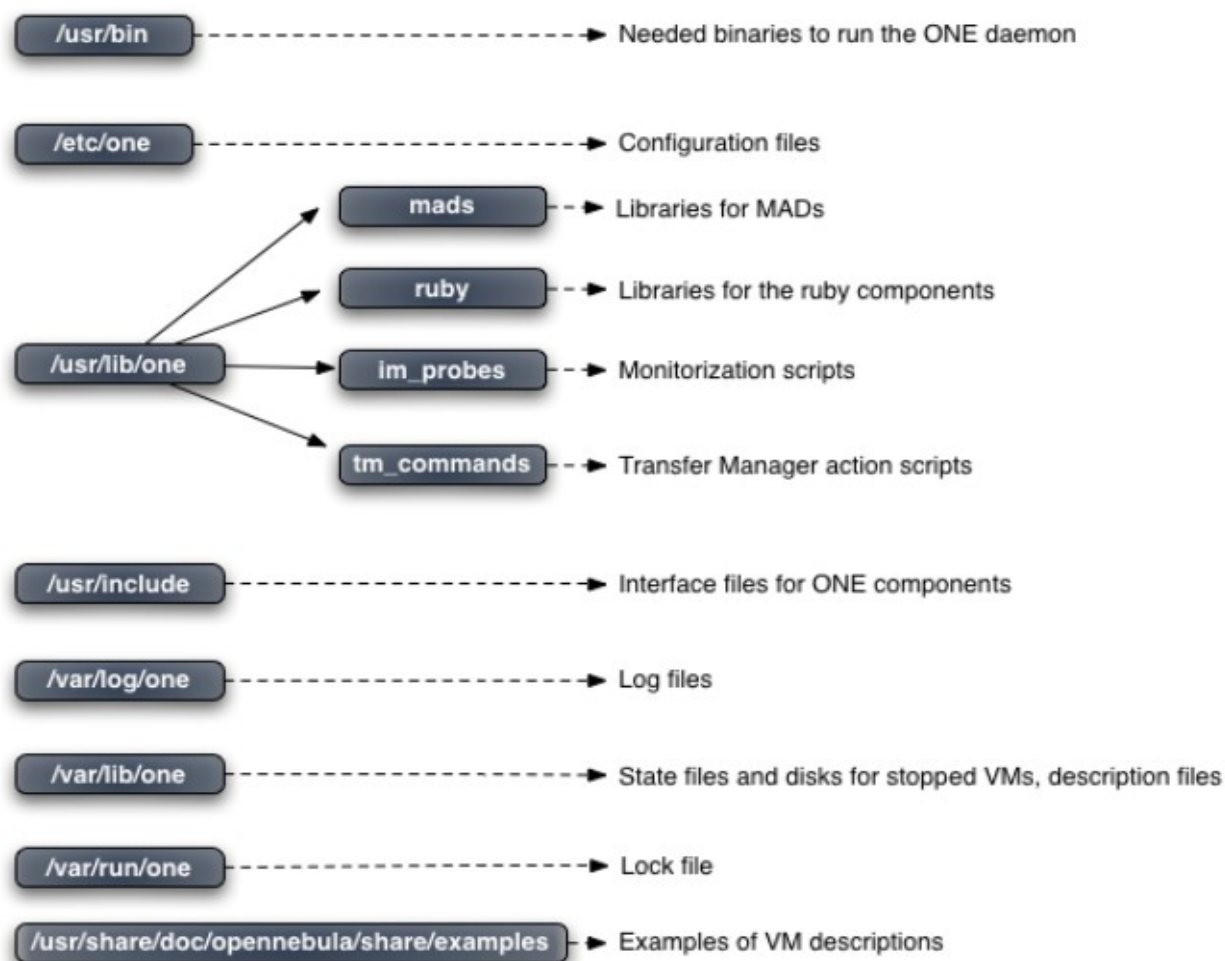
- **Front-End**, on s'executen els serveis que controlen els processos i serveis del cluster
- **Nodes**, els hipervisors habiliten als hosts per proveir als servidors virtuals de tots els recursos necessaris.
- **Repositori d'imatges** que seran la base dels servidors virtuals o guests.
- **OpenNebula daemon**, és el nucli del sistema. Gestiona el cicle de vida dels guests i orquestra els subsistemes del cluster (xarxa, emmagatzematge i hipervisors o Hosts)
- **Drivers**, programari que usa el daemon de opennebula per a gestionar els hipervisors entre d'altres recursos.
- **Usuaris :**
 - Usuari Oneadmin, és l'usuari administrador del Cloud privat i que té permisos per a realitzar qualsevol operació sobre la xarxa, Hosts i/o Guests, etc...

- Usuaris, permeten donar permisos per a fer tasques concretes al Cloud Privat.

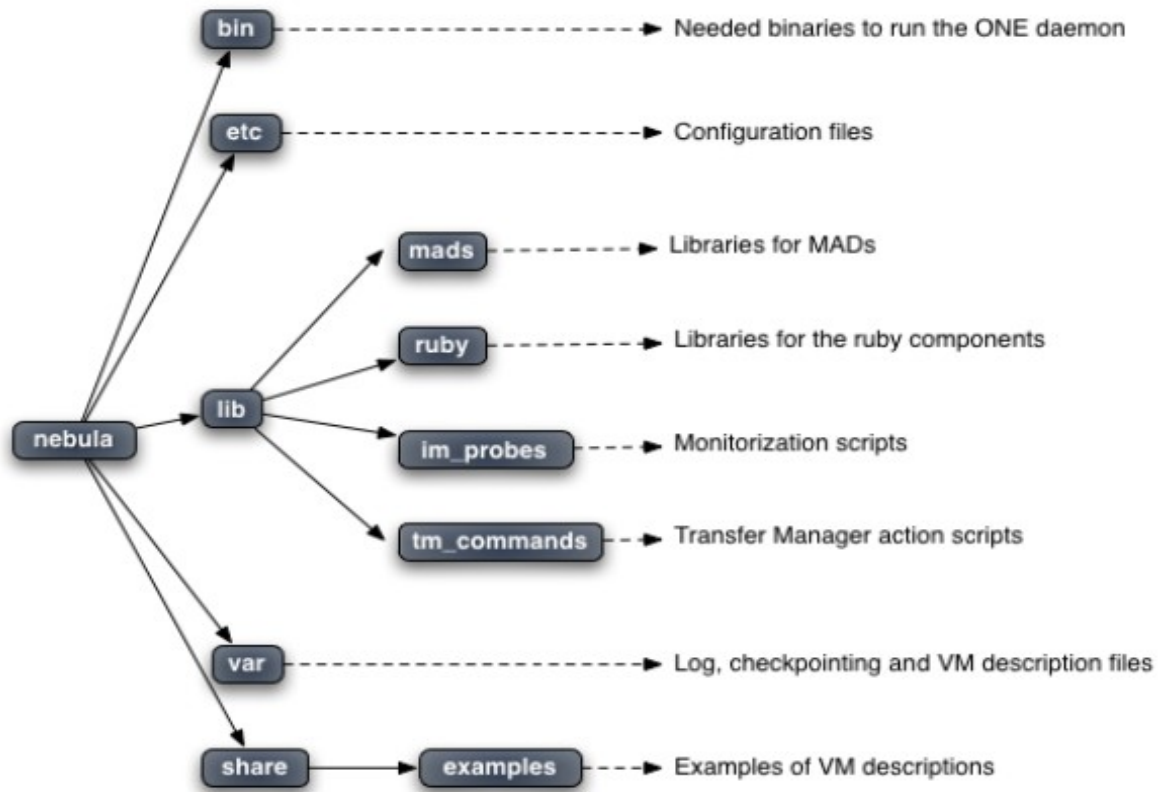
9.1.1.4.- Processos i directoris

Hi ha 2 opcions de fer la instal·lació del producte a partir del codi font o a partir dels binaris. Els 2 modes s'anomenen :

- **System-wide**, binaris, fitxers de log i configuració es guarden als directoris estàndard UNIX. Cal permisos de root per dur a terme aquesta instal·lació.



- **Self-contained**, el fitxers de OpenNebula es guarden a partir de la ruta especificada a la instal·lació, per exemple /opennebula

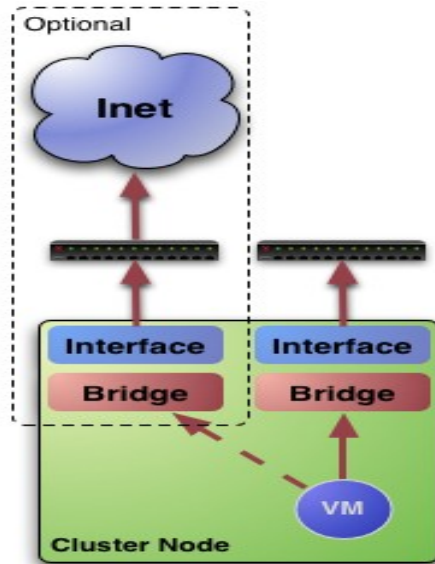


Opennebula fa ús de 3 processos principals :

- **Opennebula daemon.** Gestiona la operació i el control de tots el mòduls i el cicle de vida de les màquines virtuals.
- **Drivers.** Gestiona els hipervisors (VMM driver), la monitorització (IM Drivers) i la transferència de les màquines virtuals (TM driver).
- **Planificador.** Gestiona el desplegament de les màquines virtuals als nodes del cluster opennebula.

9.1.1.5.- Xarxes

La configuració de la xarxa no cal cap requeriment especial, opennebula crear un bridge per defecte tal i com es mostra a la imatge :



```
[root@server01 srv]# brctl show
bridge name      bridge id          STP      enabled
interfaces
virbr0           8000.000000000000  yes
xenbr0           8000.fefffffffffff no
peth0
```

Tenim la opció de crear xarxes privades virtuals i associar-les a màquines virtuals. Per a definir xarxes virtuals, cal crear una plantilla on especifiquem si es tracta d'una xarxa 'fixed' o 'ranged'.

- **Fixed** : especifica que certes Ips estan associades al servers virtuals per les seves MAC address.
- **Ranged** : similar a la configuració d'un DHCP, on l'adreça base de la IP està especificada i tota la resta d'adreces Ips són continuació

d'aquesta.

Exemple d'una plantilla per a crear una xarxa virtual



```
NAME = "Xarxa proves TFC"
TYPE = RANGED          interfaces
BRIDGE = virbr0
NETWORK_SIZE = C
NETWORK_ADDRESS = 192.168.0.0
```

Creació d'una xarxa virtual amb la plantilla creada anteriorment i amb comandes de Opennebula.



```
onevnet create network.template
```

```
onevnet list
```

ID	USER	NAME	TYPE	BRIDGE	P	#LEASES
0	oneadmin	PrivateCloud	Fixed	eth0		

9.1.1.1.- Storage

Amb Opennebula disposem de diferents mètodes per a compartir el repositori d'imatges i els directoris per les màquines virtuals:

- **Shared-FS.** En aquest cas, un sistema de fitxer del FrontEnd es exportat a la resta de nodes com mostra la imatge:



- **Non-Shared-SSH.** En aquest cas no es comparteix cap sistema de fitxers entre el front-end i els nodes. La pega és que no es pot fer 'live migration'. Per tant, no és la millor opció. Així els estats de les màquines virtuals només podran ser cloning i/o saving.
- **LVM.** Opennebula dona suport a Logical Volume Manager i podem fer ús d'una característica d'aquest, snapshotting. En aquest cas el driver de la transferència assumeix que els block devices definits a la plantilla de la màquina virtual es troben disponibles també en el node.

9.1.1.2.- Cicles de vida

9.1.1.2.1.- Imatges virtuals

Els possibles estats per a les imatges virtuals són els següents :

- **INIT.** Imatge en inicialització.
- **READY.** Imatge preparada pel seu ús.
- **USED.** Imatge que s'està fent servir per una o més màquines virtuals.
- **DISABLED.** Imatge que no es pot usar.

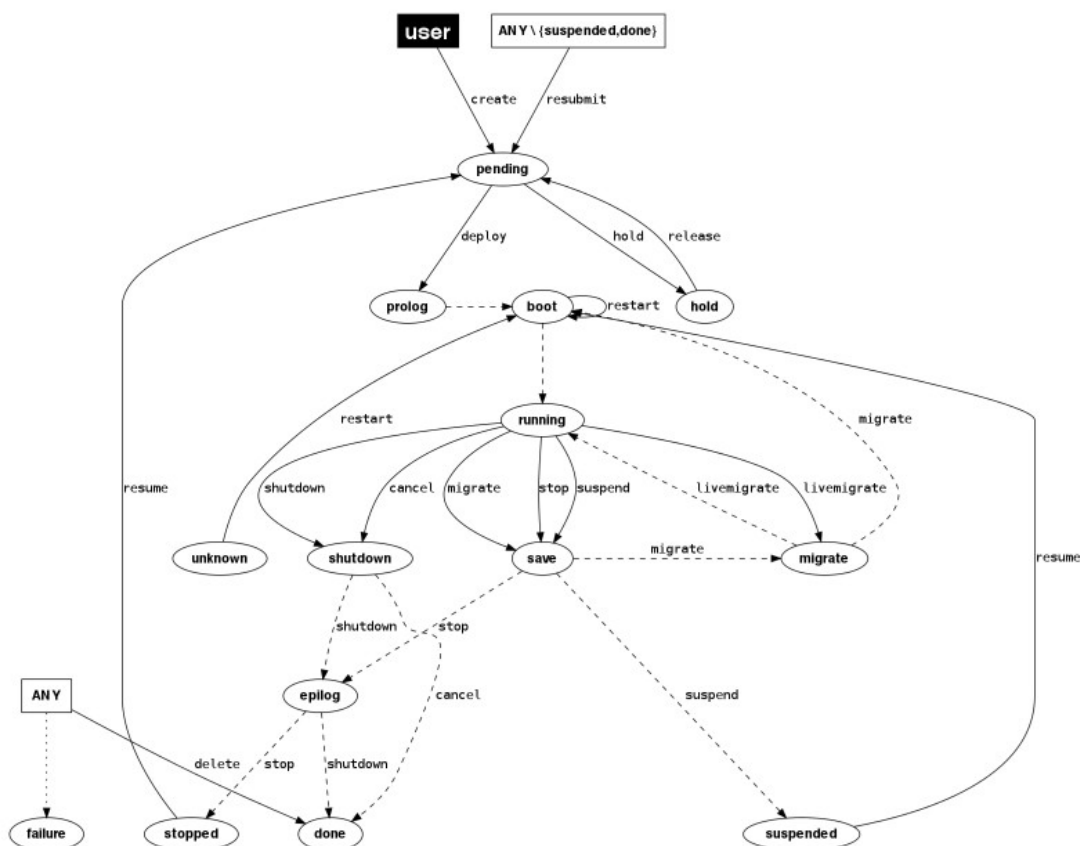
9.1.1.2.2.- Màquines virtuals

Els possibles estats per a les màquines virtuals són els següents :

- **Pending** (Pend). VM pendent que el planificador li assigni un node on executar-se.
- **Hold.** Usuari ha retingut la imatge, no pot ser desplegada.
- **Prolog.** VM està sent transferida al node.
- **Running.** VM es troba en execució al node corresponent.
- **Migrate.** VM està sent migrada a un altre node.

- **Epilog.** El sistema està fent neteja al node per ubicar la VM.
- **Stopped.** VM s'ha guardat a disc i està sent transferida al servidor.
- **Suspended.** Igual que el estat stopped, però la imatge queda al node.
- **Failed.** VM desplegada amb problemes.
- **Unknown.** VM en estat desconegut.
- **Done.** Tot i que amb la comanda onevm list no apareix, la informació de la VM queda guardada a la base de dades opennebula.
- **Boot.** VM iniciant-se al node corresponent.
- **Save.** VM guardada a disc.
- **Shutdown.** VM aturada.

La següent imatge representa de forma gràfica que els diferents estats i transicions pels que pots estar i passar una màquina virtual en opennebula:



9.1.1.3.- Fitxers

9.1.1.3.1.- Configuració

Depenent del tipus d'instal·lació realitzada, trobarem el fitxer principal de configuració a una o altre ubicació. El fitxers de configuració de opennebula els trobarem sota el directori per defecte /etc/one i el fitxer principal és oned.conf. En Aquest fitxer especificarem el setup inicial i haurem de comentar o eliminar les diferents possibilitats que ens ofereix. Per exemple, si fem ús de del hipervisor KVM, només caldria eliminar els “#” (parrilles) que apareixen al inicia de la secció VM_MAD. D'aquesta forma activem o desactivem els drivers que ens interessin.

La secció IM_MAD relaciona la informació relativa al driver per a la monitorització.

La secció VM_MAD relaciona la informació relativa al driver disponibles per negociar amb el hipervisor.

```
#-----  
# KVM Information Driver Manager Configuration  
#   -r number of retries when monitoring a host  
#   -t number of threads, i.e. number of hosts monitored at the same  
time  
#-----  
IM_MAD = [  
    name      = "im_kvm",  
    executable = "one_im_ssh",  
    arguments  = "-r 0 -t 15 kvm" ]  
#-----  
#-----  
# KVM Information Driver Manager Configuration
```



```

#   -r number of retries when monitoring a host
#   -t number of threads, i.e. number of hosts monitored at the same
time
#-----
IM_MAD = [
    name      = "im_kvm",
    executable = "one_im_ssh",
    arguments  = "-r 0 -t 15 kvm" ]
#-----

```

9.1.1.1.1.- Logs

Una part important durant el setup i quan el cloud es troba en problemes són els fitxers de logs que podem trobar al directori `/var/adm/one/oned.log`.

9.1.1.2.- Instal·lació Front-End i WorkNodes

A continuació mostrem les passes que s'han seguit per a fer una instal·lació de Opennebula private cloud a partir dels packages precompilats sobre 2 servidors Linux Ubuntu Server 11.04. Els servidors Linux implemente la tecnologia de virtualització mitjançant KVM.

- Servidor FrontEnd + WorkNode (server01) :
 - Model : HP Proliant DL385 G2
 - Specs : 2 CPU Dual-Core Opteron 2210 HE (4 cores)
- Servidor WorkNode (server02) :

- Model : HP Proliant DL385 G2
- Specs : 2 CPU Dual-Core Opteron 2210 HE (4 cores)

Instal·lació de FrontEnd/WorkNode (server01)



La instal·lació del FrontEnd la fem amb la comanda *apt-get* i instal·lant el paquet *opennebula* a partir dels repositoris oficials de Ubuntu. Aquesta acció ens instal·la el dimoni d'opennebula, **oned**, que s'encarrega de gestionar tots els serveis d'Opennebula, instal·la els prerequisits necessaris com i crea l'**usuari oneadmin** amb el home */var/lib/one*, que és l'encarregat de gestionar els diferents processos d'opennebula. La instal·lació d'aquest paquet també crea un parell de claus pública/privada SSH de tipus RSA, aquestes claus es creuen amb els WorkNodes i permeten la comunicació sense password del FrontEnd amb els WorkNodes que formen el cluster.



```
root@server01:~# apt-get install opennebula  
  
root@server01:~# apt-get install opennebula-node
```

La comanda **onehost** executada al FrontEnd ens permet realitzar qualsevol operació al host. La sintaxi és la següent:

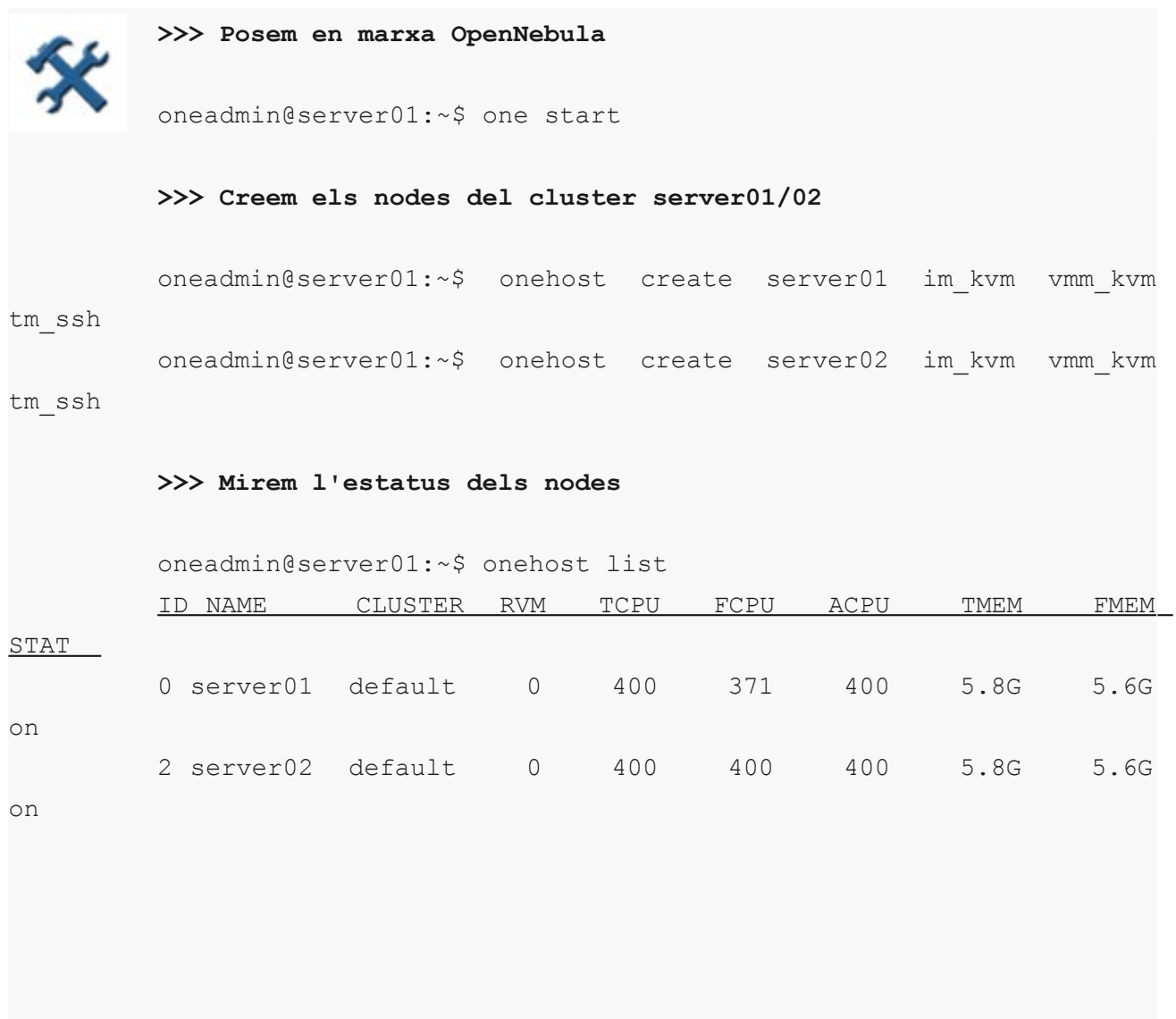
```
onehost [opcions] comandes [paràmetres]
```

Arranquem opennebula amb l'usuari oneadmin, creem els nodes del cluster (server01 i server02) i veiem el seu estatus (nodes i cluster). Abans de posar en marxa opennebula, cal editar el fitxer de configuració

`/etc/one/oned.conf` per habilitat els drivers que ens permetran crear els servidors virtuals i la transferència d'imatges entre nodes del cluster. Aquests drivers els fem servir durant la creació dels nodes del cluster, ja que, opennebula pot fer ús de diferents drivers per accedir als diferents hipervisors que podem configurar.

Afegim un node físic a OpenNebula amb la següent informació :

- **Hostname** del node del cluster o IP
- **Information driver** a fer servir per monitoritzar el host
- **Storage driver** per a clonar, copiar, moure o copiar imatges al host.
- **Virtualization driver** pel boot, aturada, arranc o migració VMs



```
>>> Posem en marxa OpenNebula

oneadmin@server01:~$ one start

>>> Creem els nodes del cluster server01/02

oneadmin@server01:~$ onehost create server01 im_kvm vmm_kvm
tm_ssh
oneadmin@server01:~$ onehost create server02 im_kvm vmm_kvm
tm_ssh

>>> Mirem l'estatus dels nodes

oneadmin@server01:~$ onehost list
```

ID	NAME	CLUSTER	RVM	TCPU	FCPU	ACPU	TMEM	FMEM
0	server01	default	0	400	371	400	5.8G	5.6G
2	server02	default	0	400	400	400	5.8G	5.6G

>>> Mirem l'estatus del cluster que es crea per defecte

```
oneadmin@server01:~$ onecluster list
  ID      NAME
  --      -
0  default
```

A continuació creem la xarxa virtual de tipus C, 192.168.1.0, amb la plantilla i la comanda següent :



>>> Creació de la plantilla

```
oneadmin@server01:~/vnet$ cat net.template
NAME = "Xarxa-TFC"
TYPE = RANGED
BRIDGE = virbr0
NETWORK_SIZE = C
NETWORK_ADDRESS = 192.168.1.0
```

>>> Creació de la xarxa virtual

```
oneadmin@server01:~/vnet$ onevnet create net.template
```

>>> Veure les xarxes creades

```
oneadmin@server01:~/vnet$ onevnet list
  ID USER      NAME                TYPE BRIDGE P #LEASES
  -- --      -
0  oneadmin Xarxa-TFC          Ranged virbr0 N          0
```

Per a la creació de servers virtuals de tipus KVM seguim les següents

passes:

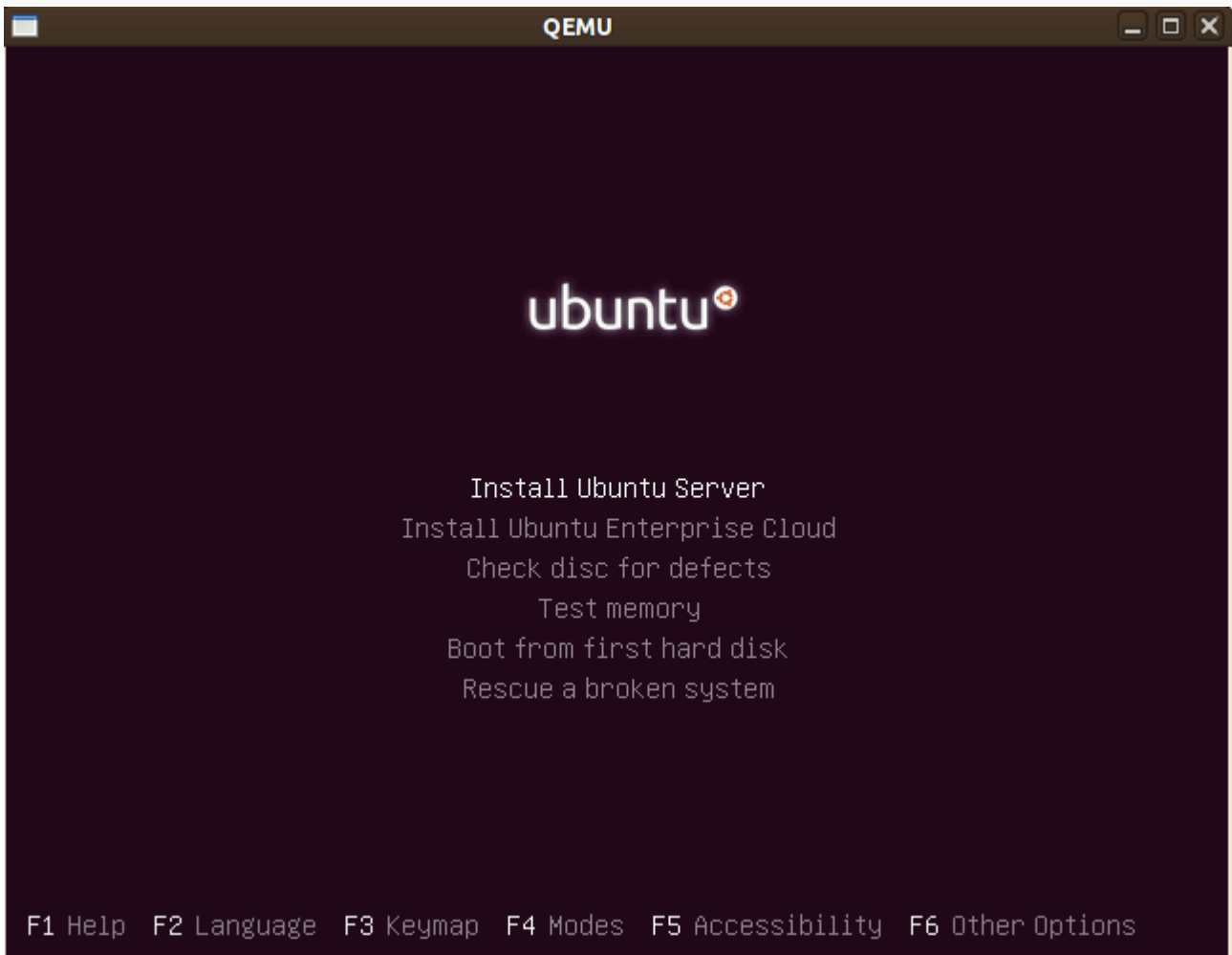


>>> Creació disk file de 4Gb a directori VM_DIR de oned.conf

```
oneadmin@server01:~/vm$ dd if=/dev/zero of=ubuntu.img bs=1M  
count=4096
```

>>> Arranquem la instal·lació des de imatge ISO

```
oneadmin@server01:~/vm$ kvm -m 512 -cdrom images/ubuntu.iso  
-boot d ubuntu.img
```



Iniciem el procés d'instal·lació.

```
oneadmin@server01:~/vm$ onevm create ubuntu.template
```

```
oneadmin@server01:~/vm$ onevm list
```

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
0	oneadmin	vm-ubunt	pend	0	0K		00 00:00:10

```
oneadmin@server01:~/vm$ onevm deploy vm-ubuntu server02
```

```
oneadmin@server01:~/vm$ onevm list
```

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
0	oneadmin	vm-ubunt	prol	0	0K	server02	00 00:03:33

```
oneadmin@server01:~/vm$ onevm top
```

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
0	oneadmin	vm-ubunt	boot	0	0K	server02	00 00:09:12

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
2	oneadmin	vm-ubunt	runn	0	0K	server02	00 00:08:35



Instal·lació de WorkNode (server02)

La instal·lació del WorkNode la fem amb la comanda *apt-get* i instal·lant el paquet *opennebula-node* a partir dels repositoris oficials de Ubuntu. Aquest package instal·la els packages i prerequisits necessaris i crea el usuari **oneadmin** i les claus SSH com en la instal·lació del FrontEnd.

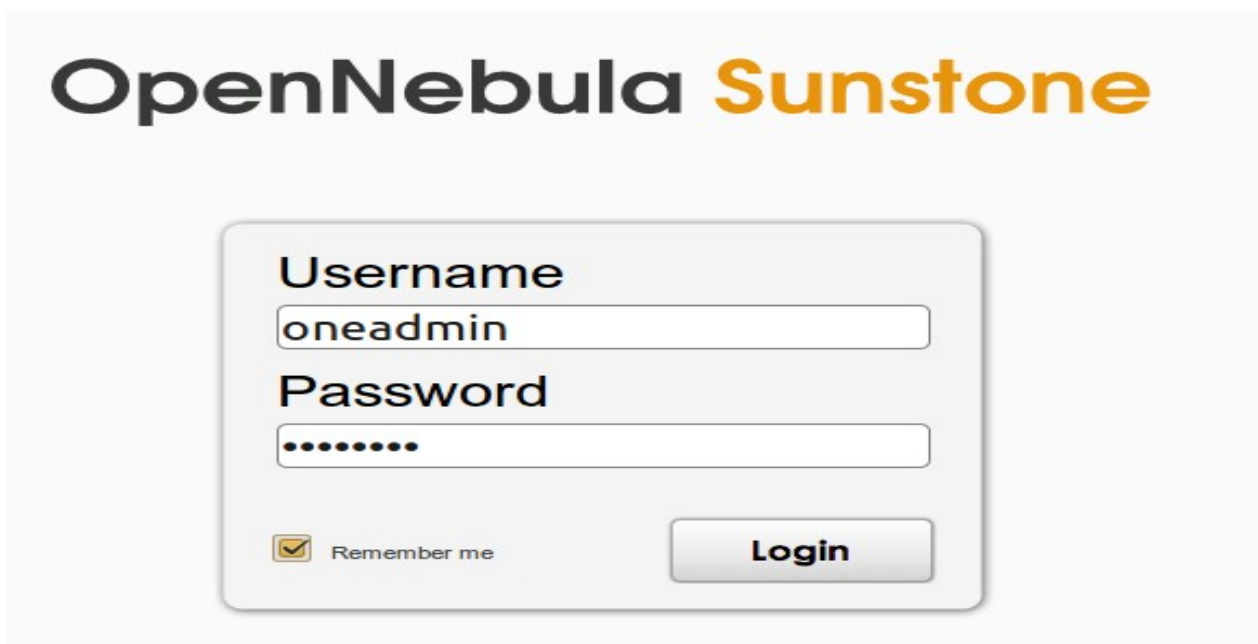
```
root@server01:~# apt-get install opennebula-node
```

9.1.1.1.- Sunstone: Cloud Operations Center

Opennebula sunstone és una eina gràfica per usuaris i administradors que simplifica les operacions de management en clouds privats i híbrids. Ens facilita la gestió física i virtual dels recursos del cloud.

Per posar en marxa opennebula sunstone disposem de la comanda 'sunstone-server' amb diferents paràmetres de port i ip per posar en marxa l'aplicació web.

La següent captura mostra com és la pantalla de presentació de sunstone: <http://192.168.0.1:4567> [oneadmin/oneadmin]



The image shows the login interface for OpenNebula Sunstone. At the top, the text 'OpenNebula Sunstone' is displayed, with 'Sunstone' in orange. Below this is a login form with the following elements:

- A 'Username' label above a text input field containing 'oneadmin'.
- A 'Password' label above a text input field filled with dots.
- A checked checkbox labeled 'Remember me'.
- A 'Login' button.

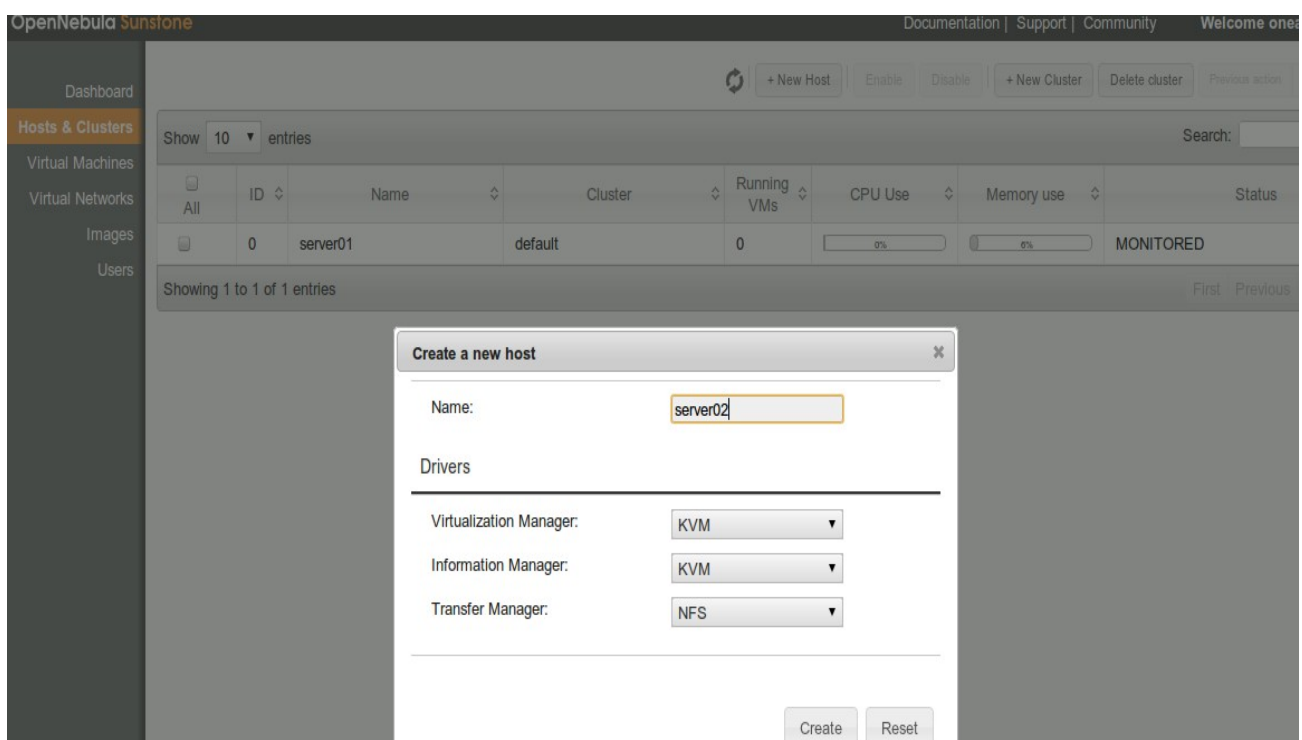
Un cop dins del aplicatiu web disposem de 6 punts importants :

- **Dashboard** : visió global de la situació del cloud
- **Hosts/Clusters** : gestió dels Hosts que pertanyen al Cloud

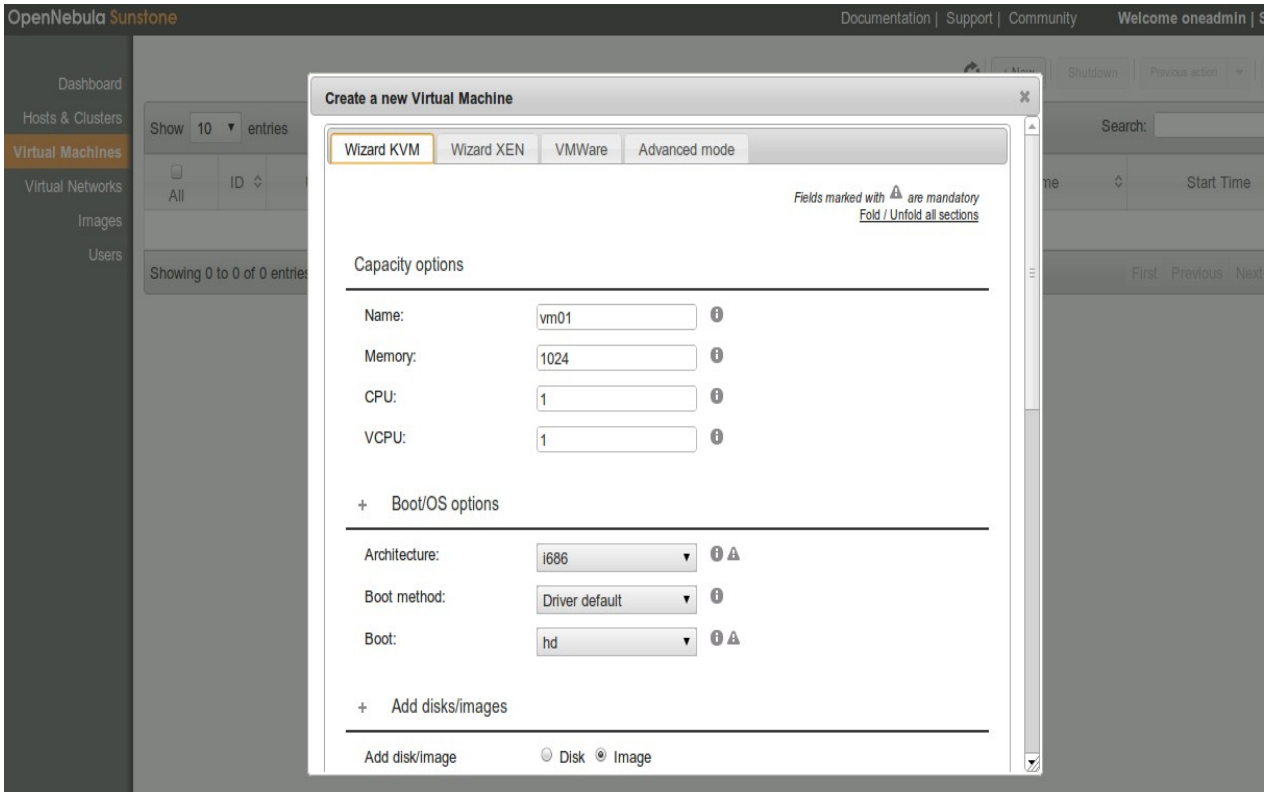
- **Virtual Machines** : gestió de les màquines virtuals creades al cloud
- **Virtual networks** : gestió de les xarxes virtuals creades al cloud
- **Images** : gestió de les imatges configurades al cloud per a poder instal·lar
- **Users** : gestió de usuaris per sunstone

Exemples des de OpenNebula SunStone:

- Creació d'un nou HOST al cluster



- Afegir servidor virtual:



9.1.2.- OpenStack

9.1.2.1.- Característiques

OpenStack és un conjunt de tecnologies open-source que faciliten la implantació d'una arquitectura cloud fàcilment escalable per clouds de tipus public i privat. Les principals característiques que ofereix OpenStack són:

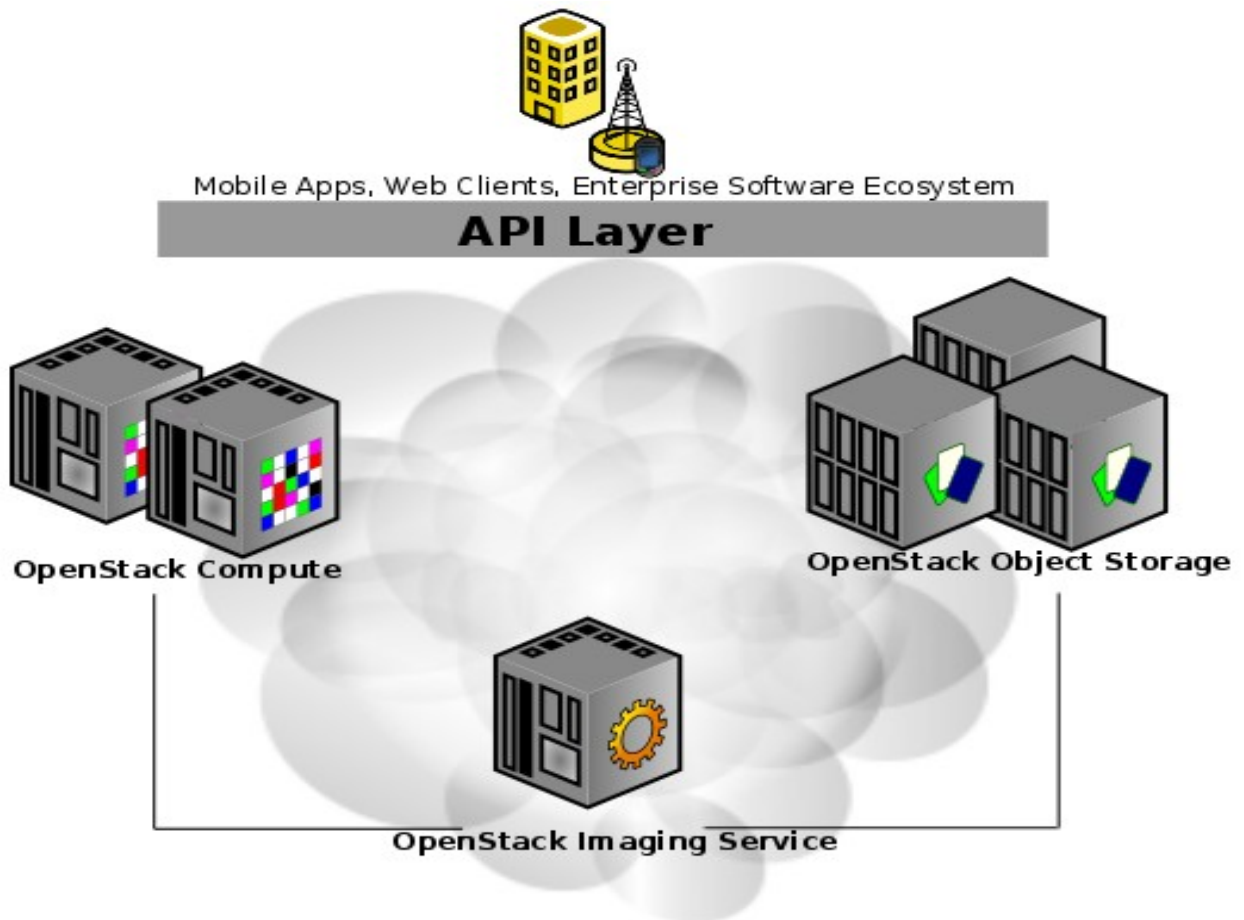
- On-demand Self-service, els usuaris poden facilitar servidors i xarxes amb una petita intervenció
- Resource Pooling, múltiples usuaris poden accedir als clouds en funció de la demanda
- Elasticitat, facilita el ràpid i escalable accés als recursos sota demanda
- Servei mesurat, els clouds optimitzen l'ús de recursos i controlen en nivell de servei ja sigui per emmagatzematge o recursos de processadors.

Així el lema principal quan treballes amb OpenStack és simple : els clouds han de ser simples d'implementar i massivament escalables.

9.1.2.2.- Arquitectura

Hi ha tres components fonamentals quan parlem de OpenStack:

- **OpenStack Compute** (codename Nova), que gestiona les màquines virtual i les xarxes virtuals.
- **OpenStack Object Storage** (condename Swift), per gestionar l'emmagatzematge, i,
- **OpenStack Image Service** (codename Glance), encarregat de gestionar les imatges.



9.1.2.3.- OpenStack Compute

OpenStack Compute, també conegut amb el nom de Nova, posa a la nostra disposició les eines per a la construcció d'una infraestructura com a servei (IaaS), la gestió d'instàncies i xarxes, i el control de l'accés al cloud des de la definició d'usuaris i projectes.

9.1.2.3.1.1.- Hipervisors

OpenStack Compute no inclou cap software per a la virtualització, però podem definir el driver que interactua amb els mecanismes de virtualització que estiguin configurats al sistema operatiu, a més d'exposar les seves funcionalitats sobre una API basada en web.

OpenStack requereix d'un hipervisor que controla a través d'una API, els hipervisors soportats són els següents :

- Hyper-V 2008
- KVM - Kernel-based Virtual Machine
- LXC - Linux Containers
- QEMU - Quick Emulator
- User-mode Linux
- VMWare - ESX/ESXi 4.1 update 1
- Xen - XenServer 5.5

9.1.2.3.1.2.- Usuaris i projectes

OpenStack Compute està dissenyat per donar servei a diferents clients/customers fent ús d'un accés basat en rols. Els rols controlen les accions que l'usuari té permès executar. A diferència d'altres IaaS, OpenStack introdueix el concepte de rol de projecte, on els recursos estan organitzats en projectes format una estructura organitzativa. Els recursos inclosos el projecte són VLANs, instàncies, imatges, claus i usuaris.

Disposem de rols següents :

- **Administrador cloud** (admin). Rol global. Usuaris amb accés complet al cloud.
- **IT Security** (itsec). Rol global. Rol per a tasques de auditoria i seguretat.
- **Project Manager** (projectmanager). Rol projecte. Permet afegir usuaris a un projecte, gestionar les imatges d'un projecte i executar i finalitzar instàncies.
- **Administrador de xarxa** (netadmin). Rol projecte. Els usuaris amb aquest rol poden assignar adreçament IP, crear i modificar regles de firewall.
- **Desenvolupador** (developer). Rol projecte. Rol de propòsit general assignat als usuaris per defecte.

9.1.2.3.1.3.- Imatges i instàncies

Una imatge és un fitxer que conté informació sobre un disc virtual que arreplega tota la informació sobre una màquina virtual en un punt del temps incloent informació del sistema operatiu i dels sistemes de fitxers.

OpenStack Compute fa ús de les comandes euca2tools per afegir o eliminar imatges.

Hi ha dos mètodes per a gestionar les imatges, a través de OpenStack Image Service (Glance), o des de OpenStack Compute (Nova). Amb Glance, se serveix una imatge al host i Nova arranca la imatge des del host on s'ha desplegat. Si la imatge és servida per Nova, aquest servei desplega i posa en marxa la imatge.

Podem entendre les imatges com a màquines virtuals funcionant sota el paraigües d'un cloud, que té un cicle de vida controlat per OpenStack Compute, responsable de construir les imatges, executar-les, reportar l'estat de les mateixes, afegir storage o finalitzar-les.

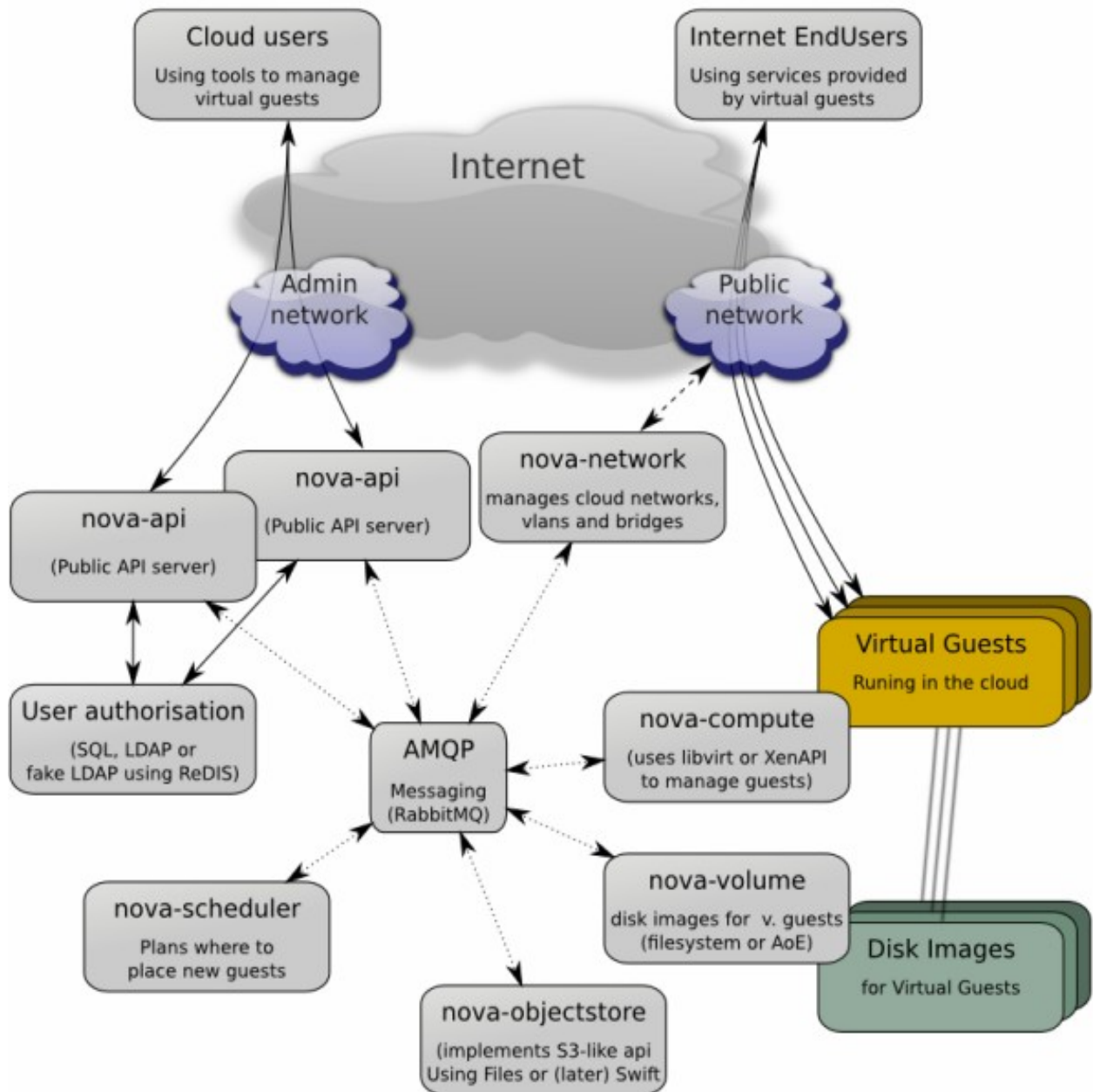
9.1.2.3.1.4.- Arquitectura interna

OpenStack Compute consisteix d'alguns components principals. Un **"controlador del cloud"** (cloud controller) conté la majoria d'aquests components i representa l'estat global que interactua amb la resta de components. Una API funciona com un front end basat en web services per interactua amb el "controlador del cloud". El **controlador de servidors** (compute controller), proveeix els recursos pels servidors i el **Object Store** que facilita els serveis d'emmagatzematge. Per conjuntar aquests mòduls o components el **auth manager** gestiona els serveis per accedir als diferents mòduls autenticats. El **volume controller**, ens permet un ràpid i permanent block-level storage pels servidors. Un **controlador de xarxa** que ens facilita l'accés a les xarxes virtuals i permet als servers comunicar-se entre ells per la xarxa pública. Finalment un **planificador** (scheduler), s'encarrega de seleccionar el compute controller més adient per desplegar una imatge.

OpenStack Compute està construït sota una arquitectura basada en la missatgeria. La majoria de components poder funcionar en múltiples servidors incloent compute controller, volume controller, network controller, i

object store. El controlador del cloud es comunica amb el object store via HTTP (Hyper Text Transfer Protocol) i amb la resta de components via AMQP (Advanced Message Queue Protocol).

La següent imatge il·lustra la arquitectura interna.



9.1.2.3.1.5.- Storage

Un 'volum' és un dispositiu que podem connectar i desconnectar a una imatge com si d'una unitat USB es tractés. Només poden estar assignats a una única imatge, així que no treballa com una SAN (Storage Area Network). Per a compartir dades entre imatges cal configurar Samba o un NFS (Network File System).

9.1.2.3.1.6.- Instal·lació OpenStack Compute

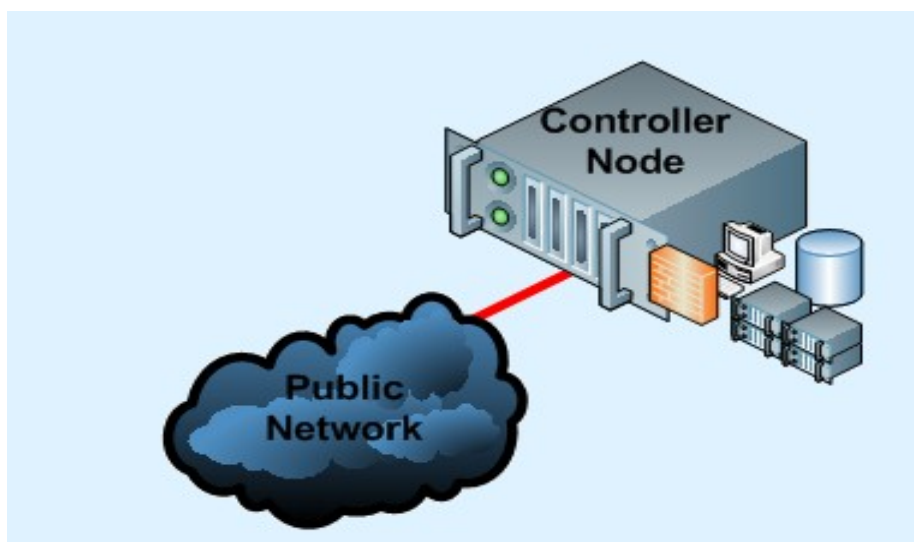
A continuació mostrem les passes que s'han seguit per a fer una instal·lació de OpenStack Compute sobre 1 servidor Linux Ubuntu Server 11.04. El servidor Linux implemente la tecnologia de virtualització mitjançant KVM. La instal·lació es basa en un únic node amb les següents característiques HP Proliant DL385 G2

hardware

Name	Speed	Cores	
Dual-Core AMD Opteron(tm) Processor 2210 HE	1000	1	
Dual-Core AMD Opteron(tm) Processor 2210 HE	1000	1	
Dual-Core AMD Opteron(tm) Processor 2210 HE	1000	1	
Dual-Core AMD Opteron(tm) Processor 2210 HE	1000	1	
Virtualization	true		
RAM Size	6271889408		
Device	Size	Used	Mount Point
/dev/cciss/c0d0	36385505280	-1	
/dev/cciss/c0d0p1	238787584	30323712	/boot
/dev/cciss/c0d0p2	1024	-1	
/dev/cciss/c0d0p5	36127637504	-1	
/dev/dm-0	34569453568	-1	
/dev/dm-1	1526726656	-1	
Interface	Type	Name	
eth0	II BCM5708 Gigabit Ethernet		
eth1	II BCM5708 Gigabit Ethernet		

Les tasques marcades seguides per a la instal·lació són:

- Setup administrador de OpenStack
- Crear un projecte OpenStack
- Generar parell de claus pública/provada
- Crear una imatge per posar en marxa en forma de màquina virtual.
- Pujar les imatges i arrancar-les.
- Imatge de la instal·lació



>>> Instal·lació de requisits RabbitMQ Messaging server
NoSQL Redis database.

```
root@server01:~# apt-get install rabbitmq-server redis-server
root@server01:~# apt-get install python-nova
root@server01:~# apt-get install python-software-properties
root@server01:~# add-apt-repository ppa:nova-core/trunk
```

>>> Instal·lació components ObjectStore, Compute, Schedule I
network.


```
root@server01:~# apt-get install -y nova-api nova-objectstore
root@server01:~# apt-get install -y nova-compute nova-scheduler
root@server01:~# apt-get install -y nova-network nova-doc
```

>>> Instal·lació euca2ools Suite

```
root@server01:~# apt-get install -y euca2ools unzip
```

>>> Instal·lació i setup MySql

```
root@server01:/tmp# export MYSQL_PASS=nova
root@server01:/tmp# export NOVA_PASS=notnova
```

```
root@server01:/tmp# cat <<MYSQL_PRESEED | debconf-set-selections
>mysql-server-5.1      mysql-server/root_password      password
                        $MYSQL_PASS
>mysql-server-5.1      mysql-server/root_password_again  password
                        $MYSQL_PASS
>mysql-server-5.1      mysql-server/start_on_boot       boolean true
MYSQL_PRESEED
```

```
root@server01:~# apt-get install -y mysql-server
root@server01:/tmp# service mysql restart
```

```
root@server01:/tmp# mysql -uroot -p$MYSQL_PASS -e 'CREATE
DATABASE nova;'
```

```
root@server01:/tmp# mysql -uroot -p$MYSQL_PASS -e "GRANT ALL
PRIVILEGES ON *.* TO 'nova'@'%' WITH GRANT OPTION;"
```

```
root@server01:/tmp# mysql -uroot -p$MYSQL_PASS -e "SET PASSWORD
FOR 'nova'@'%' = PASSWORD('$NOVA_PASS');" 
```

>>> Setup compute-network

```
root@server01:~# vi /etc/network/interfaces
# Networking for OpenStack Compute
auto br100
iface br100 inet static
```

```
address 192.168.0.101
bridge_ports      eth1
bridge_stp        off
bridge_maxwait    0
bridge_fd         0
```

```
root@server01:~# /etc/init.d/networking restart
root@server01:~# restart libvirt-bin; restart nova-network;
root@server01:~# restart nova-compute; restart nova-scheduler
root@server01:~# restart nova-api; restart nova-objectstore;
```

>>> Estat dels serveis

```
root@server01:~# /var/lib/nova/bin/nova-manage service list
nova-controller nova-scheduler enabled  :-) 2011-06-13 10:00:38
nova-controller nova-network enabled   :-) 2011-06-13 10:00:41
nova-controller nova-volume enabled   :-) 2011-06-13 10:00:41
nova-controller nova-compute enabled   :-) 2011-06-13 10:00:41
```

>>> Setup fitxer conf nova.conf

```
root@server01:/etc/nova# cat nova.conf
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--verbose
--daemonize=1
--s3_host=192.168.0.100
--rabbit_host=192.168.0.100
--ec2_api=192.168.0.100
--ec2_url=http://192.168.0.100:8773/services/Cloud
--fixed_range=192.168.0.0/12
--network_size=8
--routing_source_ip=192.168.0.100
--sql_connection=mysql://nova:notnova@192.168.0.100/nova

root@server01:/etc/nova# addgroup nova
root@server01:/etc/nova# chown -R root:nova /etc/nova
```

```

root@server01:/etc/nova# chmod 640 /etc/nova/nova.conf
>>> Creació del usuari administrador i assignació rol

root@server01:~/bin/nova-manage user create jclaret
export EC2_ACCESS_KEY=97692049-5ba5-416a-8337-2e5cb3ebbee2
export EC2_SECRET_KEY=ea3386b9-331d-4bf3-85ec-debdce111140

root@server01:/nova-manage role add jclaret cloudadmin

>>> Creació d'un Nova Project

root@server01:~/bin/nova-manage project create nubeblog jclaret

>>> Creació dels certificats i parell de claus pel projecte

root@server01:~/bin/nova-manage project zipfile nubeblog jclaret

root@server01:~/bin/nova-manage project zipfile nubeblog jclaret
root@server01:# unzip /root/creds/novacreds.zip -d /root/creds/
Archive: /root/creds/novacreds.zip
  extracting: /root/creds/novarc      << variables entorn
  extracting: /root/creds/pk.pem     << Clau privada
  extracting: /root/creds/cert.pem   << Certificat
  extracting: /root/creds/cacert.pem << CA certificat

>>> Configuració variables entorn

root@server01:/etc/nova# cat /root/creds/novarc >> ~/.bashrc
root@server01:/etc/nova# source ~/.bashrc

root@server01:/etc/nova# euca-authorize -P tcp -p 22 default

>>> Importem imatges a Nova-computer

root@server01:~/bin/nova-manage image="ttylinux-uec-amd64-12.tar.gz"
root@server01:~/bin/nova-manage image="ttylinux-uec-amd64-12.tar.gz"
root@server01:~/bin/nova-manage image="ttylinux-uec-amd64-12.tar.gz"
root@server01:~/bin/nova-manage image="ttylinux-uec-amd64-12.tar.gz"
root@server01:~/bin/nova-manage image="ttylinux-uec-amd64-12.tar.gz"
Mon Jun 13 12:28:08 CEST 2011: ===== extracting image =====
kernel : ttylinux-uec-amd64-12.1_2.6.35-22_1-vmlinuz
ramdisk: ttylinux-uec-amd64-12.1_2.6.35-22_1-initrd
image   : ttylinux-uec-amd64-12.1_2.6.35-22_1.img

```

```
Mon Jun 13 12:28:08 CEST 2011: ===== bundle/upload kernel =====
Mon Jun 13 12:28:10 CEST 2011: ===== bundle/upload ramdisk =====
Mon Jun 13 12:28:13 CEST 2011: ===== bundle/upload image =====
Mon Jun 13 12:28:16 CEST 2011: ===== done =====
emi="ami-6e59e2c4"; eri="ari-36e6ef4f"; eki="aki-7181295a";
```

>>> Instàncies

```
root@server01:~# euca-add-keypair nubeblog > nubeblog.pem
root@server01:~# chmod 600 nubeblog.pem
root@server01:~# euca-run-instances -k nubeblog -t m1.tiny ami-6e59e2c4
```

```
RESERVATION r-4gglj3jp          nubeblog          default
INSTANCE i-00000001          ami-6e59e2c4      scheduling
```

```
root@server01:~# euca-describe-instances
```

```
RESERVATION r-4gglj3jp          nubeblog default
INSTANCE i-00000001 ami-6e59e2c4      10.0.0.2 running
```

>>> Permisos per accedir per SSH a la instància

```
root@server01:~# euca-authorize default -P tcp -p 22 -s 0.0.0.0/0
root@server01:~# euca-authorize default -P icmp -t -1:-1
```

>>> Accés a la instància

```
root@server01:~# ssh 10.0.0.2 -i nubeblog.pem
```

```
# uname -a
```

```
Linux i-00000001 2.6.35-22-virtual #35-Ubuntu SMP Sat Oct 16
23:19:29 UTC 2010 x86_64 GNU/Linux
```

>>> Aturada instància

```
root@server01:~# euca-terminate-instances i-00000001
```

9.1.2.3.1.7.- Fitxers configuració

Gran part de la informació de configuració de OpenStack Compute està disponible en el fitxer `/etc/nova/nova.conf`. Una possible configuració és la mostrada a continuació:

```
root@server01:/etc/nova# cat nova.conf
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--verbose
--daemonize=1
--s3_host=192.168.0.100
--rabbit_host=192.168.0.100
--ec2_api=192.168.0.100
--ec2_url=http://192.168.0.100:8773/services/Cloud
--fixed_range=192.168.0.0/12
--network_size=8
--routing_source ip=192.168.0.100
```

9.1.2.4.- OpenStack Object Storage

OpenStack Object Storage és un sistema d'emmagatzematge escalable encarregat de facilitar l'accés d'espai al cloud. No es tracta dels file systems tradicionals o dels sistemes basats en SAN o NAS. És un sistema dissenyat per ser usat per diferents clients o usuaris, on cada usuari s'autentica mitjançant el mòdul **swauth**.

9.1.2.4.1.- Objectes i contenidors

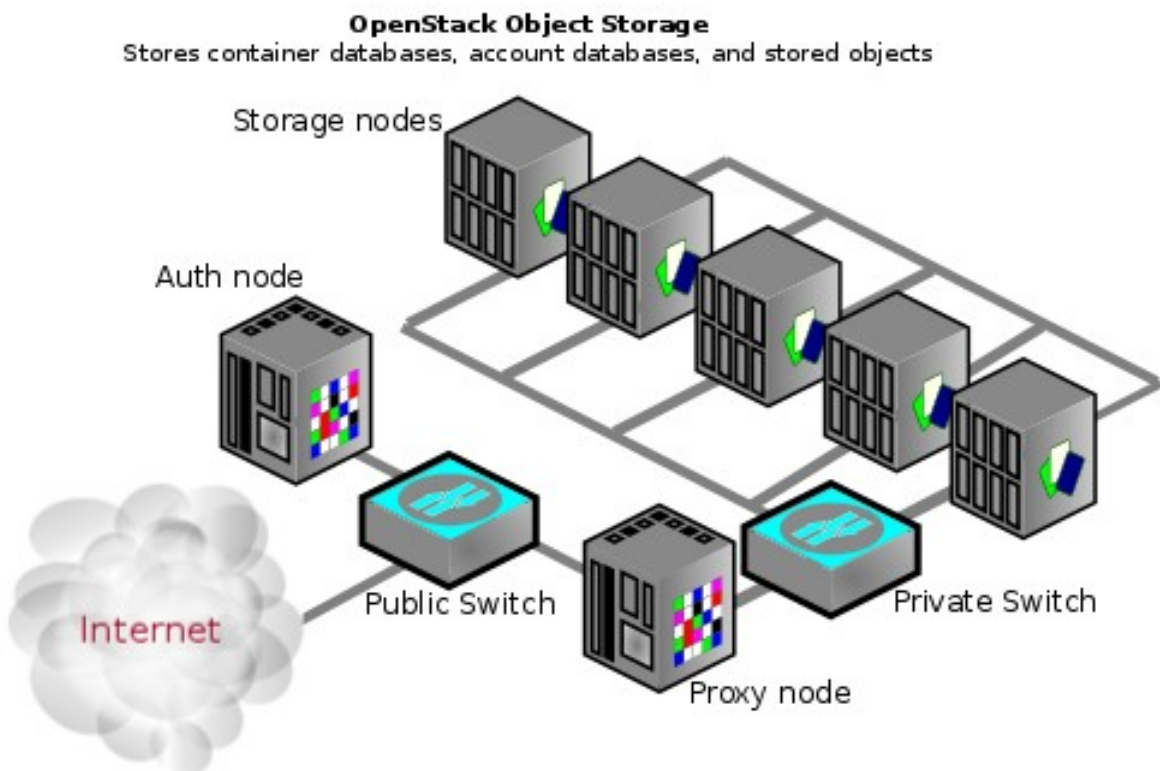
Un contenidors és un compartiment d'emmagatzematge per les dades amb l'objectiu d'organitzar les dades, podem imaginar un contenidor com una carpeta o directori que trobem als sistemes windows o unix respectivament. Amb un contenidor podem assignar un número il·limitat de contenidors als usuaris on es guardaran les dades.

Un objecte és una entitat bàsica d'emmagatzematge que forma part

d'un contenidor. Les dades es poden guardar amb diferents formats de compressió i encriptació. La mida màxim per defecte d'un objecte és de 5Gb, però si fem ús dels mòduls de large-object podrem obtenir objectes més grans.

9.1.2.4.2.- Instal·lació Swift

Tot i que la instal·lació realitzada és d'un únic node, la següent imatge il·lustra un disseny típic a RackSpace.



Les passes seguides per a la instal·lació de Openstack Swift són els següents:



>>> Instal·lació de OpenStack Storage i prerequisits

```
root@server01:~# apt-get install python-software-properties
```

```
root@server01:~# add-apt-repository ppa:swift-core/ppa
root@server01:~# apt-get update
root@server01:~# apt-get install swift
```

>>> Configuració OpenStack Storage

```
root@server01:~# mkdir -p /etc/swift
root@server01:~# chown -R swift:swift /etc/swift/
root@server01:~# vi /etc/swift/swift.conf
    [swift-hash]
    swift_hash_path_suffix = 1234567890
```

>>> Instal·lació/configuració Proxy Node

```
root@server01:~# apt-get install swift-proxy memcached
root@server01:~# cd /etc/swift
root@server01:~# openssl req -new -x509 -nodes -out cert.crt
-keyout cert.key
root@server01:~# vi /etc/memcached.conf
    -l 192.168.0.100 (<PROXY_LOCAL_NET_IP>)
root@server01:~# service memcached restart
root@server01:~# vi /etc/swift/proxy-server.conf
```

```
[DEFAULT]
# Enter these next two values if using SSL
certifications
cert_file = /etc/swift/cert.crt
key_file = /etc/swift/cert.key
bind_port = 8080
workers = 8
user = swift

[pipeline:main]
pipeline = healthcheck cache swauth proxy-server

[app:proxy-server]
use = egg:swift#proxy
allow_account_management = true

[filter:swauth]
```

```

use = egg:swift#swauth
# Highly recommended to change this.
super_admin_key = swauthkey

[filter:healthcheck]
use = egg:swift#healthcheck

[filter:cache]
use = egg:swift#memcache
memcache_servers = 192.168.0.100:11211

```

>>> Creació objectes i contenidors

```

root@server01:# cd /etc/swift
root@server01:# swift-ring-builder account.builder create 18 3 1
root@server01:# swift-ring-builder container.builder create 18 3
1
root@server01:#swift-ring-builder object.builder create 18 3 1
root@server01:~#swift-ring-builder    object.builder    add    z1-
10.0.0.1:6000/sdb1 100
root@server01:~#swift-ring-builder    container.builder  add    z1-
10.0.0.1:6001/sdb1 100
root@server01:~#swift-ring-builder    account.builder    add    z1-
10.0.0.1:6002/sdb1 100

```

>>> Validació objectes i contenidors

```

root@server01:# swift-ring-builder account.builder
account.builder, build version 1
262144 partitions, 3 replicas, 1 zones, 1 devices, 100.00 b
alance
The minimum number of hours before a partition can be reassigned
is 1
Devices: id zone ip address  port name weight partitions balance
meta
      0      1          10.0.0.1  6002      sdb1 100.00          0
-100.00

```



```

root@server01:# swift-ring-builder container.builder
container.builder, build version 1
262144 partitions, 3 replicas, 1 zones, 1 devices, 100.00
balance
The minimum number of hours before a partition can be reassigned
is 1

```

```

Devices:id zone ip address port name weight partitions balance meta
0 1 10.0.0.1 6001 sdb1 100.00 0 -100.00

```

```

root@server01:# swift-ring-builder object.builder
object.builder, build version 1
262144 partitions, 3 replicas, 1 zones, 1 devices, 100.00
balance
The minimum number of hours before a partition can be reassigned
is 1

```

```

Devices:id zone ip address port name weight partitions balance
meta
0 1 10.0.0.1 6000 sdb1 100.00 0
-100.00

```

>>> Instal·lació/configuració Storage Node

```

root@server01:# apt-get install swift-account swift-container
swift-object xfsprogs

```

```

root@server01:# vi /etc/rsyncd.conf

```

```

uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = 192.168.0.100

```

```

[account]
max connections = 2
path = /srv/node/
read only = false
lock file = /var/lock/account.lock

```

```

[container]
max connections = 2
path = /srv/node/

```

```

read only = false
lock file = /var/lock/container.lock

[object]
max connections = 2
path = /srv/node/
read only = false
lock file = /var/lock/object.lock

root@server01:# vi /etc/default/rsync
    RSYNC_ENABLE = true
root@server01:# service rsync start
root@server01:# vi /etc/swift/account-server.conf
    [DEFAULT]
    bind_ip = 192.168.0.100
    workers = 1
    [pipeline:main]
    pipeline = account-server

    [app:account-server]
    use = egg:swift#account
    [account-replicator]
    [account-auditor]
    [account-reaper]

root@server01:# vi /etc/swift/container-server.conf
    [DEFAULT]
    bind_ip = 192.168.0.100
    workers = 1
    [pipeline:main]
    pipeline = container-server
    [app:container-server]
    use = egg:swift#container
    [container-replicator]
    [container-updater]
    [container-auditor]

root@server01:# vi /etc/swift/object-server.conf
    [DEFAULT]
    bind_ip = 192.168.0.100

```

```

workers = 2
[pipeline:main]
pipeline = object-server
[app:object-server]
use = egg:swift#object
[object-replicator]
[object-updater]
[object-auditor]

root@server01:# swift-init object-server start
root@server01:# swift-init object-replicator start
root@server01:# swift-init object-updater start
root@server01:# swift-init object-auditor start
root@server01:# swift-init container-server start
root@server01:# swift-init container-replicator start
root@server01:# swift-init container-updater start
root@server01:# swift-init container-auditor start
root@server01:# swift-init account-server start
root@server01:# swift-init account-replicator start
root@server01:# swift-init account-auditor start

>>> Creació de volums i assignació a la màquina virtual
root@server01:# euca-create-volume -s2 -znova (2Gb)
root@server01:# euca-describe-volumes
VOLUME      vol-00000001      2      nova      available

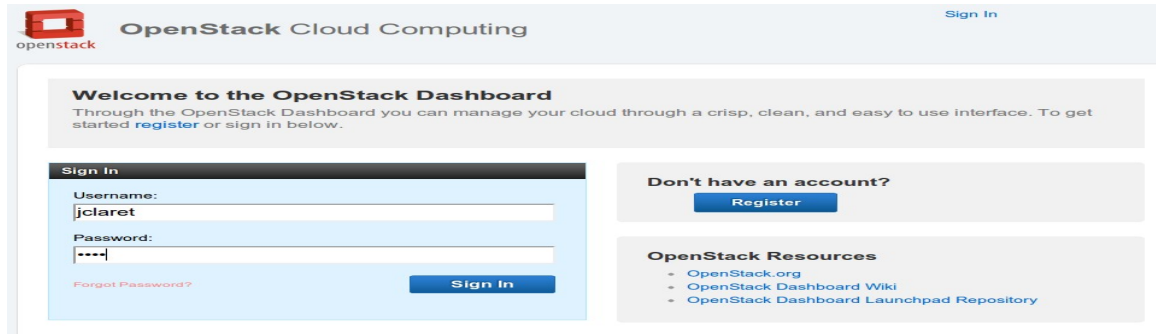
root@server01:# euca-attach-volume vol-00000001 -i i-00000001
-d /dev/vdb
root@server01:# euca-describe-volumes
VOLUME      vol-00000001      2      nova      in-use

```

9.1.2.4.3.- OpenStack Dashboard

Openstack dashboard és una eina web que ens serveix per administrar les imatges carregades, gestionar els volums, projectes, usuaris, rols, keys i volums. Les següents imatges mostren alguns exemples d'ús amb el dashboard.

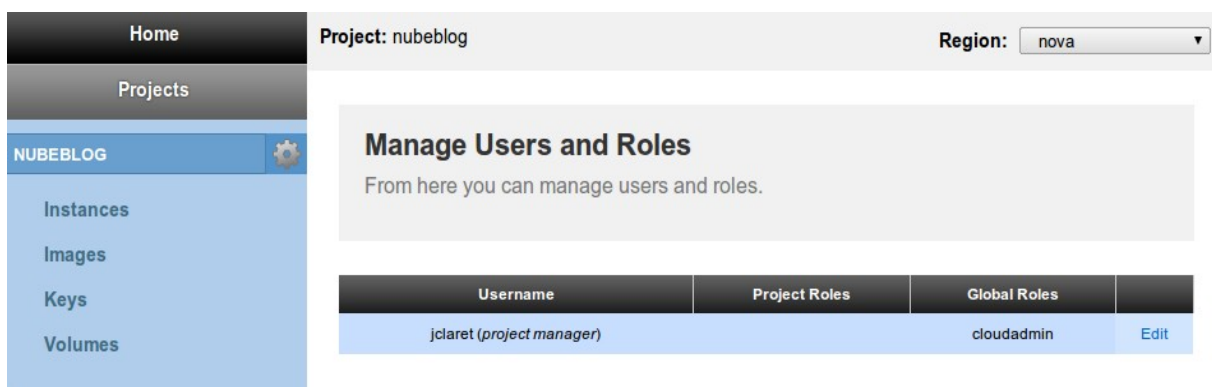
- Pàgina d'inici d'OpenStack Dashboard



- Pàgina d'inici d'OpenStack Dashboard mostrant els projectes



- Pàgina principal d'on podem gestionar instàncies, imatges, keys i volums.



- **Instàncies en funcionament.**

Project: nubeblog Region: nova

Instances

Instances are virtual servers launched from images. You can launch instances from the [images tab](#).

ID	Image	Size	IP	State
i-00000003 (Server 3)	ami-6e59e2c4	m1.medium	10.0.0.2	running

Buttons: Terminate, Show Console

- **Imatges publicades**

Project: nubeblog Region: nova

Images

Images are snapshots of running systems which can easily be deployed to run one or more instances.

Project Images

No images currently available.

OpenStack Images

No images currently available.

Community Images

ID	Description	Owner
ami-57de8016	bucket/natty-server-uec-amd64.img.manifest.xml	Launch
ami-6e59e2c4	bucket/ttylinux-uec-amd64-12.1_2.6.35-22_1.img.manifest.xml	Launch

9.1.3.- Taula Comparativa

La següent taula mostra algunes de les principals característiques que podem trobar en els 2 productes IaaS:

Classe	Funcionalitats	OpenNebula	OpenStack
Gestió Usuaris	Crear	SI	SI
	Eliminar	SI	SI
	Autenticar	SI	SI
	Modificar contrasenyes	SI	SI
	Mostrar	SI	SI
	Quotes	SI	SI
Gestió de rols d'usuaris	Administrador	SI	SI
	Convencional	SI	SI
Gestió de nodes	Crear	SI	SI
	Eliminar	SI	SI
	Activar	SI	SI
	Desactivar	SI	SI
	Mostrar	SI	SI
	Nodes Linux	SI	SI
	Nodes Windows	NO	SI
	CPU online	SI	SI
	RAM online	SI	SI
	HD online	SI	SI
Gestió de imatges	Registrar	SI	SI
	Eliminar	SI	SI
	Modificar	SI	SI
	Activar	SI	SI
	Desactivar	SI	SI
	Publicar	SI	SI
	Mostrar	SI	SI
Gestió de màquines virtuals	Iniciar	SI	SI
	Iniciar mostrant HOST	NO	SI
	Instal·lació ràpida	NO	SI
	Desplegar	SI	SI
	Apagar	SI	SI
	Migrar	SI	SI
	Migrar online	SI	SI
	Aturar	SI	SI
	Suspendre	SI	SI
	Esborrar	SI	SI
	Reiniciar	NO	SI
	Reempendre	SI	SI
	Guardar	SI	SI
	Mostrar	SI	SI
	Ejecutar Scripts	SI	SI
	CPU online	SI	SI
	MEM online	SI	SI
HD online	SI	SI	
Gestió de clusters	Definir	SI	SI
	Afegir nodes	SI	SI
	Eliminar nodes	SI	SI
	Mostrar	SI	SI
	Crear	SI	SI

Gestió de xarxes virtuals	Eliminar	SI	SI
	Mostrar	SI	SI
Gestió de contexte	Contextualització	SI	SI
	Contextualització Linux	SI	SI
	Contextualització Windows	NO	SI
Interfície d'usuari	Línea de comandes	SI	SI
	Web	SI	SI
Tipus de clouds	Privada	SI	SI
	Publica	SI	SI
	Híbrida	SI	SI
Interfície de cloud	Cloud públic	SI	SI
	Suport per Amazon EC2	SI	SI
	Suport per ElasticHosts	SI	SI
	Accés simultani a diversos clouds	SI	SI
	Interfície EC2 Query	SI	SI
	Interfície OGF OCCI	SI	SI
	Interfície vCloud	SI	SI
Hipervisors	Suport Xen	SI	SI
	Suport KVM	SI	SI
	Suport VMWare	SI	SI
	Suport VirtualBox	NO	SI
	Suport Quemu	NO	SI
	Gestió del planificador	SI	SI
Federació	Capacitat de federació	SI	SI
Abstracció	Abstracció de la infraestructura	SI	SI
	Abstracció serveis de virtualització	SI	SI
	Abstracció de storage	SI	SI
	Abstracció de la xarxa	SI	SI
	Fa ús d'estandards oberts	SI	SI
Interoperabilitat	Expansió de funcionalitats via software adicional	SI	SI
Seguretat	Comunicació interna basada en SSL	SI	SI
	Comunicació externa basada en SSL	SI	SI
	Monitorització de la xarxa	NO	SI
Gestió de la monitorització	Monitorització dels nodes	SI	SI
	Monitorització de màquines virtuals	SI	SI
	Monitorització de serveis	SI	SI

9.1.4.- Conclusions.

Si fem cas a les notícies que podem llegir a qualsevol dels principals portals d'informació tecnològica, en un curt espai de temps totes les empreses, ja siguin grans o petites, s'hauran de pujar al carro del cloud computing. Per estalviar costos, ser més eficients o simplement per no quedar fora de la nova moda tecnològica, les grans empreses estan preparant el terreny pel futur, i cada cop més veiem la paraula cloud als seus productes.

En aquest breu projecte hem observat i posat en pràctica que implantar una plataforma cloud és una possibilitat que qualsevol empresa ja té al seu abast, tot i que, hauran d'invertir en formació i temps per duu a terme grans plataformes. Hem vist que per a petites empreses, la opció més recomanable és un producte més senzill conceptualment i en la seva instal·lació, en canvi, per a empreses més grans és més recomanable OpenStack per la seva solidesa, tot i que la seva instal·lació és més complicada i cal dedicar-hi moltes més hores, al final s'aconsegueix una plataforma més estable. Tot i això, aquests dos IaaS (sobretot opennebula) han de madurar i provar-se en diferents plataformes, i sobretot, facilitar la seva instal·lació, ja que, durant les diferents instal·lacions s'ha trobat diferents problemes de incompatibilitats de paquets linux amb els packages d'instal·lació del IaaS. A pesar dels contratemps, que durant el projecte han estat molts, cal destacar la complexitat i alhora el potencial d'aquesta capa del Cloud, IaaS. El desplegament de plataformes de desenvolupament o fins i tot de producció serà possible fer-ho en uns temps rècord gràcies a les configuracions d'imatges al servidors de imatges i de les plantilles amb diferents possibilitats de configuració.

Pel que fa a la documentació que podem trobar a les webs tant de Opennebula o OpenStack va millorant amb el temps, però deixen de documentar molts aspectes.

9.1.4.1.- Disseny tècnic. Instal·lació Servidors Linux Ubuntu.

Per a dur a terme els casos pràctics disposem de 2 servidors amb els següents components:

Model :	HP Proliant DL385 G2
RAM :	6Gb
CPUs :	Dual-core (total 4 Cores).
HD :	2x36 Gb + 1x72Gb
SO Linux :	Ubuntu Server 11.04 (natty)

Els servidors HP Proliant DL385 han estat configurats i actualitzats a nivell de:

- **Firmware.**
[<http://h18004.www1.hp.com/products/servers/management/core-management-100.html>]
- **ILO** (Integrity Integrated Lights-Out). Per de gestió per accés via HTTPS o SSH. Les següents imatges mostren la pantalla de benvinguda a la ILO i un cop hem accedit, les diferents opcions d'administració que podem trobar.



Integrated Lights-Out 2

HP ProLiant



Login name:

Password:

192.168.0.205 https://192.168.0.205/ie_index.htm



Integrated Lights-Out 2

HP ProLiant

System Status Remote Console Virtual Media Power Management Administration

Status Summary

Summary

- System Information
- iLO 2 Log
- IML
- Diagnostics
- iLO 2 User Tips
- Insight Agent

Server Name: SERVER02; ProLiant DL385 G2

Serial Number / Product ID: CZC7220F9R / 434940-421

UUID: 39343334-3034-5A43-4337-323230463952

System ROM: A09 07/11/2009; backup system ROM: 01/08/2007

System Health: ✔ Ok

Internal Health LED: ✔ Ok

Server Power: ✔ ON

UID Light: OFF

Last Used Remote Console: Remote Console

Latest IML Entry: POST Error: 1720-S.M.A.R.T. Hard Drive Detects Imminent Failure

iLO 2 Name: SERVER02

iLO 2 FQDN: SERVER02.

License Type: iLO 2 Advanced

iLO 2 Firmware Version: 2.05 12/16/2010

IP address: 192.168.0.205

Active Sessions: iLO 2 user:admin

Latest iLO 2 Event Log Entry: Browser login: admin - 192.168.0.201(DNS name not found).

- PSP. Instal·lació del Proliant Support Pack
[<http://h18000.www1.hp.com/products/servers/management/psp/index.html>]

10.- Bibliografia.

- <http://www.opennebula.org/documentation:rel2.2:features>
- <http://en.wikipedia.org/wiki/OpenNebula>
- http://en.wikipedia.org/wiki/Infrastructure_as_a_service#Infrastructure
- [http://en.wikipedia.org/wiki/Eucalyptus_\(computing\)](http://en.wikipedia.org/wiki/Eucalyptus_(computing))
- <http://www.linux-kvm.org/>
- <http://libvirt.org>
- <http://linux.die.net/man/1/virsh>
- www.centos.org
- http://www.cesga.es/component/option,com_docman/task,doc_download/gid,580/Itemid,13/lang,es/
- <http://www.rpath.com/corp>
- <http://www.salesforce.com/es/cloudcomputing/>
- <http://www.salesforce.com/es/cloudcomputing/>
- <http://wiki.centos.org/HowTos/KVM#head-6cbcdf8f149ebcf19d53199a30eb053a9fc482db>
- <http://centos.karan.org/>
- <http://www.openstack.org/>

- <http://www.openstack.org/>
- <http://www.ubuntu.com/>
- <https://wiki.ubuntu.com/DocumentationTeam>
- http://twitter.com/#!/opnstk_com_mgr
- http://twitter.com/#!/opnstk_com_mgr
- <https://help.ubuntu.com/community/UEC/Images>
- <http://uec-images.ubuntu.com/releases/> (the images for use in UEC or Eucalyptus)
- <https://uksysadmin.wordpress.com/2011/03/03/openstack-nova-centos-instance/>
- <http://links.visibli.com/links/f76224>
- <http://cloudcomputingexpo.com/>
- http://blogs.forrester.com/stefan_ried/10-07-06-forresters_cloud_computing_taxonomy
- <http://pyarra.blogspot.com/2011/03/ubuntu-server-on-hp-proliant-managing.html>

