

## Treball final de grau

**Estudi: Grau en Enginyeria Informàtica**

**Títol: Instal·lació i configuració d'un servei de cloud en una aula informàtica**

**Document: Memòria**

**Alumne: SANTI PÉREZ CARMONA**

**Director/tutor:** Pere Vilà Talleda

**Departament:** Arquitectura i Tecnologia de Computadors

**Àrea:** Arquitectura i Tecnologia de Computadors

**Convocatòria (mes/any): Juny 2015**

# ÍNDEX

<b>1. INTRODUCCIÓ, MOTIVACIONS, PROPÒSIT I OBJECTIUS DEL TREBALL.....</b>	<b>7</b>
1.1. INTRODUCCIÓ .....	7
1.2 MOTIVACIONS .....	8
1.3 PROPÒSIT .....	9
1.4 OBJECTIUS .....	9
<b>2. ESTUDI DE VIABILITAT .....</b>	<b>10</b>
<b>3. METODOLOGIA.....</b>	<b>13</b>
<b>4. PLANIFICACIÓ .....</b>	<b>15</b>
<b>5. MARC DE TREBALL I CONCEPTES PREVIS.....</b>	<b>17</b>
5.1 CLOUD COMPUTING.....	17
5.2 MÀQUINA VIRTUAL .....	19
5.3 HYPERVISOR .....	19
5.4 INSTÀNCIA.....	19
5.5 IMATGE .....	19
5.6 API REST .....	19
5.7 CLOUD MANAGEMENT PLATFORM.....	20
5.8 CMPS DE CODI LLIURE .....	20
5.8.1 <i>OpenStack</i> .....	21
Característiques principals .....	21
Eines disponibles .....	22
Requisits del sistema .....	22
5.8.2 <i>OpenNebula</i> .....	22
Característiques principals .....	23
Requisits del sistema .....	23
5.8.3 <i>Eucalyptus</i> .....	23
Característiques principals .....	24
Eines disponibles .....	24
Requisits del sistema .....	25
Computacionals .....	25
Emmagatzematge i memòria .....	25
Xarxa .....	25
5.9 WAKE ON LAN (WOL).....	26
5.10 DEEPFREEZE .....	26
5.11 VMWARE ESXI .....	26
5.12 VMWARE PLAYER .....	26
5.14 VMWARE VIX .....	26
<b>6. REQUISITS DEL SISTEMA.....</b>	<b>27</b>
6.1 REQUISITS OPENSTACK .....	27
6.2 REQUISITS SERVIDOR D'AULA .....	28
6.3 REQUISITS ORDINADOR DE TREBALL .....	28
<b>7. ESTUDIS I DECISIONS.....</b>	<b>29</b>
7.1 PLATAFORMA D'ADMINISTRACIÓ DEL CLOUD .....	29
7.2 ESCENARIS .....	31
7.2.1 <i>ESXi</i> .....	31

Maquinari .....	31
Programari.....	32
7.2.2 <i>Exportació del cloud a l'aula</i> .....	32
Maquinari .....	32
Programari.....	32
7.2.3 <i>Escenari final</i> .....	33
Maquinari .....	33
Programari.....	34
7.3 SERVEIS DE L'OPENSTACK.....	34
7.3.1 <i>Identity service (Keystone)</i> .....	35
Funcions del servei .....	35
Definicions per entendre l'Identity.....	35
7.3.2 <i>Image service (Glance)</i> .....	37
Components del servei .....	37
7.3.3 <i>Compute service (Nova)</i> .....	38
Àrees del servei: .....	38
API:.....	38
Core (nucli) de computació:.....	39
Xarxa per les VMs: .....	39
Interfície a través de la consola .....	39
Administració d'imatges (escenari EC2) .....	39
Clients des de línia de comandes i altres interfícies .....	40
Altres components.....	40
7.3.4 <i>OpenStack networking (Neutron)</i> .....	41
Components .....	41
Servidor.....	41
Plugins i agents de xarxa de l'Openstack .....	41
Cua de missatges .....	42
Conceptes .....	42
7.3.5 <i>Dashboard (Horizon)</i> .....	43
7.3.6 <i>Orchestration module (Heat)</i> .....	44
7.3.7 <i>Telemetry module (Ceilometer)</i> .....	45
7.3.8 <i>Data processing service (DPS - Sahara)</i> .....	46
<b>8. ANÀLISI I DISSENY DEL SISTEMA .....</b>	<b>48</b>
8.1 ESCENARI ESXI.....	48
8.2 ESCENARI PC PERSONAL.....	49
8.3 ESCENARI FINAL.....	50
<b>9. IMPLEMENTACIÓ I PROVES .....</b>	<b>54</b>
9.1 ESCENARI ESXI.....	54
9.1.2 PREPARACIÓ DEL SOFTWARE BÀSIC DE LES VMs .....	63
<i>Network Time Protocol (NTP)</i> .....	63
Network i Compute1: .....	63
<i>Definició del repositori i instal·lació de paquets</i> .....	64
<i>Base de dades</i> .....	64
<i>Servei de missatgeria (Message broker, intercanviador de missatges)</i> .....	66
9.1.2 INSTAL·LACIÓ I CONFIGURACIÓ DELS SERVEIS OPENSTACK .....	68
9.1.2.1 <i>Identity service (Keystone)</i> .....	68
Instal·lació i configuració .....	68
Prerequisits.....	68
Instal·lació i configuració .....	68

Finalització de la instal·lació .....	71
Creació de contenidors ( <i>tenants</i> ), usuaris i rols .....	71
Prerequisits (variables d'entorn) .....	72
Creació .....	72
Creació de l'entitat de servei i de l'API .....	73
Prerequisits .....	73
Creació .....	74
Verificacions .....	75
Creació dels <i>scripts</i> de l'entorn client de l'OpenStack .....	76
9.1.2.2 <i>Image service (Glance)</i> .....	79
Instal·lació i configuració .....	79
Prerequisits .....	79
Instal·lar i configurar els components del servei .....	80
Finalització de la instal·lació .....	83
Comprovacions .....	83
9.1.2.3 <i>Compute service (Nova)</i> .....	87
Instal·lació i configuració del node Controller .....	87
Prerequisits .....	87
Instal·lació i configuració del software específic del servei nova .....	88
Finalitzar la instal·lació .....	90
Instal·lació i configuració d'un node de Computació .....	90
Instal·lació i configuració dels components de l'hypervisor al node Compute1 .....	90
Finalització de la instal·lació .....	92
Comprovacions .....	92
9.1.2.4 <i>Networking component</i> .....	95
Instal·lació i configuració al node Controller .....	95
Prerequisits .....	95
Instal·lació i configuració del software específic del servei neutron .....	96
Configurar el plugin Modular Layer 2 (ML2) .....	98
Configurar el Compute (nova) perquè utilitzi el Networking (neutron) .....	99
Finalització de la instal·lació .....	100
Verificacions .....	101
Instal·lació i configuració del node Network .....	101
Prerequisits .....	101
Instal·lació dels paquets específics de Networking (neutron) .....	102
Configuració dels components del Networking (neutron) .....	102
Configurar el plugin Modular Layer 2 (ML2) .....	103
Configurar l'agent Layer-3 .....	105
Configurar l'agent DHCP .....	105
Configurar l'agent de metadades .....	107
Configurar el servei Open vSwitch (OVS) .....	108
Finalització de la instal·lació .....	109
Verificacions .....	109
Instal·lació i configuració del node Compute1 .....	109
Prerequisits .....	109
Instal·lació dels components del Networking (neutron) .....	110
Configurar els components comuns del Networking (neutron) .....	110
Configurar el plugin Modular Layer 2 (ML2) .....	111
Configurar el servei Open vSwitch (OVS) .....	113
Configurar el servei Compute (nova) perquè utilitzi el Networking (neutron) .....	113
Finalització de la instal·lació .....	113
Verificacions .....	114
Creació de les xarxes inicials .....	116
Xarxa External .....	116
Crear la xarxa External .....	116



Crear una subxarxa a la xarxa External .....	117
Xarxa tenant .....	117
Crear la xarxa tenant.....	118
Crear una subxarxa a la xarxa Tenant .....	118
Crear un router .....	119
Verificacions de connectivitat.....	119
9.1.2.5 <i>Dashboard (Horizon)</i> .....	121
Requeriments del sistema .....	121
Instal·lació i configuració .....	122
Instal·lació.....	122
Configuració del Dashboard .....	122
Finalització de la instal·lació .....	122
Verificacions .....	123
9.1.2.6 <i>Orchestration module (Heat)</i> .....	125
Instal·lació i configuració .....	125
Prerequisits.....	125
Instal·lació dels components del Heat.....	127
Finalització de la instal·lació .....	129
Verificacions .....	130
9.1.2.7 <i>Telemetry module (Ceilometer)</i> .....	134
Instal·lació i configuració al node Controller .....	134
Prerequisits.....	134
Instal·lar i configurar els paquets del mòdul Telemetry .....	136
Finalització de la instal·lació .....	138
Instal·lació i configuració dels nodes de Computació.....	139
Instal·lar i configurar l'agent.....	139
Configurar les notificacions .....	140
Finalització de la instal·lació .....	141
Configurar el servei d'Image (glance) .....	141
Verificacions .....	142
9.1.2.8 <i>Data processing service (DPS - Sahara)</i> .....	143
Instal·lació i configuració al node Controller .....	143
Prerequisits.....	143
Instal·lar i configurar els components dels servei.....	145
Comprovacions.....	150
9.1.3 CONJUNT DE SERVEIS.....	150
9.2 ESCENARI PC PERSONAL.....	152
9.2.1 EXTRACCIÓ DE LES VMS DE L'ESXI .....	152
9.2.2 PROCÉS D'ADAPTACIÓ A L'ESCENARI .....	156
<i>Disminuir la memòria disponible .....</i>	156
<i>Creació de les xarxes Management, Tunnel i External .....</i>	158
<i>Verificacions.....</i>	163
9.3 ESCENARI FINAL: AULA III-04i.....	167
9.3.1 SERVIDOR D'AULA .....	167
<i>Tasques i eines .....</i>	167
<i>Cloud Control .....</i>	169
9.3.2 ESTACIONS .....	174
<i>Habilitació del Wake on Lan .....</i>	175
<i>Habilitació de l'execució remota de comandes (Pstools).....</i>	179
<i>Habilitació de l'VT-x.....</i>	179
<i>Script d'adaptació de les màquines virtuals .....</i>	182
<b>10. IMPLANTACIÓ I RESULTATS.....</b>	<b>185</b>

<b>11. CONCLUSIONS .....</b>	<b>191</b>
<b>12. TREBALL FUTUR .....</b>	<b>192</b>
<b>13. BIBLIOGRAFIA .....</b>	<b>194</b>
<b>14. ANNEXOS .....</b>	<b>197</b>
14.1 ARRENCADA / PARADA DEL CLOUD .....	197
<i>Fitxer computeNodes.txt</i> .....	197
<i>Fitxer controllerNode.txt</i> .....	197
<i>Fitxer hosts.txt</i> .....	197
<i>Fitxer networkNode.txt</i> .....	197
<i>Fitxer start.bat</i> .....	197
<i>Fitxer startLog.bat</i> .....	199
<i>Fitxer stop.bat</i> .....	199
<i>Fitxer stopLog.bat</i> .....	200
14.2 FITXERS D'SCRIPT D'ADAPTACIÓ DELS NODES.....	200
<i>Fitxer hosts</i> .....	200
<i>Fitxer setNodeParams.sh</i> .....	201
14.3 FITXER D'SCRIPT CLOUDACCESS.BAT .....	203
<b>15. MANUAL D'USUARI I/O INSTAL·LACIÓ.....</b>	<b>204</b>

# 1. Introducció, motivacions, propòsit i objectius del treball

## 1.1. Introducció

En els últims anys, el *Cloud Computing* està despertant cada cop més interès tan al món empresarial com pels usuaris finals, però, què és realment? Es tracta d'un conjunt de serveis accessibles a través de la xarxa (ja sigui Internet o una xarxa privada) que un conjunt de proveïdors ofereixen, ja que disposen d'una sèrie de recursos informàtics compartits (xarxes, servidors, emmagatzemament, aplicacions i serveis) per poder fer-ho. A més, aquests recursos s'han d'oferir als seus consumidors de manera transparent, sense complicacions i sense la intervenció directa del proveïdor.

Si ens ho mirem des del punt de vista dels usuaris, aquests no s'han de preocupar de com està muntada la infraestructura que ofereix aquests serveis ni de la seva gestió; són característiques que han de ser transparents per ells. L'única cosa que necessiten per ser-ne beneficiaris és tenir un dispositiu que es pugui connectar a una xarxa que els comuniqui amb el *cloud*, a més de les credencials necessàries per poder fer servir aquests recursos (normalment usuari i contrasenya).

Tot i això, abans que es comencés a parlar de "la computació al núvol" i dels serveis que s'oferixen a través del núvol, ja existien serveis que s'oferien a través de la xarxa. Així doncs, què és el que els diferencia? Es tracta de tres característiques principals, que són:

- Es venen segons la demanda, típicament per temps i/o trànsit.
- Són elàstics: els usuaris poden decidir la quantitat de recursos que necessiten en el moment que vulguin.
- El proveïdor s'encarrega de tota la gestió tècnica.

Tot i ser un concepte força nou, a l'any 1999 l'empresa Salesforce.com va començar a oferir les seves aplicacions a través d'una web (*Software-as-a-Service*), i es podria dir que van ser els pioners en aquest camp. Poc temps després, a l'any 2002 va entrar en joc el gegant Amazon, el qual, com moltes altres companyies, durant la bombolla de les "empreses.com" havia redimensionat a l'engròs els seus recursos TI, fent-ne servir llavors un 10% tan sols.

Veient el desaprofitament de tota aquesta infraestructura, els responsables d'Amazon van decidir que s'havia de trobar alguna manera de treure benefici de tots aquests recursos infrautilitzats. En conseqüència, en primer lloc van oferir un servei d'emmagatzematge al núvol (anomenat S3) i poc temps després de computació (anomenat EC2), sent tots dos, com es pot observar, serveis d'infraestructura (*Infrastructure-as-a-Service*).

Doncs bé, aquesta idea de fer servir els recursos infrautilitzats per treure'n algun profit, és la causa de la realització d'aquest treball. Més concretament, el laboratori de xarxes es tracta d'una aula - la 04i, per més detall - de l'edifici PIII de la Universitat que disposa de 18 màquines amb una gran capacitat computacional. Fins al moment,

només es feien servir exclusivament per la docència, quedant la major part del temps aturades, i en conseqüència, desaprofitades.

Veient això, el temps d'inactivitat de l'aula podria ser aprofitat per la realització de càlculs que presentin un alt cost tan a nivell temporal com de recursos necessaris, és a dir, HPC (*High Performance Computing*). Dit d'una altra manera, es tractaria d'oferir els recursos de l'aula com un servei d'infraestructura, tal i com se'ls va acudir a Amazon.

## 1.2 Motivacions

Des que vaig descobrir que em volia dedicar a la informàtica, sempre m'havia interessat, per sobre dels diversos camps que engloben aquesta disciplina, el d'administració de sistemes. El fet de poder controlar un conjunt de màquines, programar les tasques que aquestes han de realitzar, controlar-ne les comunicacions... tot això (i més coses) és el que més m'agrada d'aquest fantàstic món.

Tan és així que la meva experiència professional relacionada amb la informàtica és en el món TIC, concretament com a tècnic de camp i administrador de sistemes. Va ser durant 6 anys i em vaig dedicar a l'administració i reparació de servidors, estacions, comunicacions, *software* i *hardware*, així com la resolució d'averies de tots aquests elements, també.

Una altra motivació ha estat el creixement exponencial del món del *cloud* i la previsió de futur que se'n fa, que és la de seguir en la mateixa direcció. De fet, durant els anys com a tècnic, vaig poder observar l'evolució ascendent d'aquest nou paradigma: d'una banda, la meua empresa, en els seus correus d'informació corporativa, cada cop feia més èmfasi sobre el món *cloud*; d'altra banda, però, també vaig poder veure els seus efectes negatius sobre els tècnics de camp.

Més concretament, aquests efectes van en el sentit en què les estacions de treball s'estaven transformant en simples terminals; els processos de més importància s'estaven traslladant cada cop més al núvol, provocant que l'estació anés perdés més rellevància, fent que el seu cost de manteniment fos més elevat que simplement canviar la màquina.

El món de les TIC evoluciona cap a la virtualització de recursos, el *cloud* i l'automatització de processos de manera remota, característiques que reuneix el treball en sí. Aquesta sembla ser l'aposta de futur, així que crec que adquirir aquests coneixements m'ajudarà enormement en la meua carrera professional.

Un altre factor que m'interessa de manera notable és el món de la virtualització. En el curs de diverses assignatures al llarg de la carrera, hem fet servir aquestes eines per la realització de pràctiques (*VirtualBox* i *VMware*), facilitant-nos fer muntatges de sistemes que, si els haguéssim hagut de muntar amb màquines físiques, hagués estat impossible. En el món laboral també vaig poder veure que aquesta eina es fa servir cada cop més, com per exemple, en la virtualització de servidors, enlloc de comprar noves màquines físiques.

### 1.3 Propòsit

El propòsit d'aquest treball consisteix en conferir a l'aula esmentada a la introducció (04i del PIII), una reconversió que li permeti assolir el funcionament de dos rols independents entre sí, que són:

- Aula docent normal: amb noves facilitats per realitzar el seu manteniment i actualitzacions d'una manera centralitzada i automatitzada.
- Cloud o Servei de càlcul: que permeti usar els recursos de l'aula durant les nits i els caps de setmana per realitzar experiments i càlculs.

### 1.4 Objectius

A continuació s'enumeraran els objectius que s'havien d'assolir per poder dur a terme els propòsits que s'han explicat a l'apartat anterior:

- Estudiar el sistema que hi havia muntat a l'aula en termes de maquinari, programari instal·lat i comunicacions.
- Explorar les diferents alternatives que existeixen per la creació i administració de *clouds* i valorar quina s'adapta millor a la funcionalitat requerida.
- Dissenyar una implementació del servei de *cloud* de manera que aquesta preservi la independència entre les funcionalitats de l'aula (com a docent i com a centre de càlcul).
- Disseny i implementació d'un escenari equivalent a l'aula per poder fer les proves necessàries.
- Idear un sistema que s'encarregui de la gestió i administració automatitzada de l'aula, incloent els seus dos futurs rols.
- Implantació del *cloud* i del sistema de gestió i administració a l'aula.

## 2. Estudi de viabilitat

El principal paràmetre que calia analitzar per determinar la viabilitat del projecte era aconseguir que el rol de *cloud* de l'aula no afectés de cap manera al correcte funcionament d'aquesta com a aula docent normal, i viceversa. Per saber si ambdós funcions eren compatibles entre elles, calia veure si existia alguna combinació de *software* que ho permetés.

Així, en primer lloc calia saber com estava muntada l'aula, i en segon lloc els requisits que necessitava un *cloud* per poder funcionar. Aleshores, d'una banda, tenim que l'aula d'informàtica disposa de 18 màquines amb el sistema operatiu Windows 7 i, d'altra banda, tenim que la infraestructura interna d'un *cloud* es tracta d'una sèrie d'ordinadors en xarxa que:

- El seu sistema operatiu permet virtualitzar els seus propis recursos *hardware*. Per realitzar aquesta tasca, tan el Windows com el Linux ho permeten fer.
- Disposen d'un *software* que permet administrar d'una manera transparent aquests recursos *hardware*. El programari d'administració de *clouds* que he fet servir, l'*OpenStack*, necessita que les màquines que l'executen tinguin com a sistema operatiu el Linux. Aquesta característica va fer que es plantegessin una sèrie de decisions que s'havien de prendre a l'hora de realitzar la implementació a l'aula.

Per tant, l'escenari que tenim és:

- En primer lloc, els ordinadors de l'aula que han d'executar el *cloud* tenen com a sistema operatiu el Windows.
- En segon lloc, els requisits per operar un *cloud* diuen que es necessita un sistema operatiu Linux per poder fer-ho.

Aquests dos punts fan que la única manera de fer que l'aula adquireixi els dos rols, *cloud* i aula docent, passi per fer que els dos sistemes operatius coexisteixin, per la qual cosa hi havia dues opcions:

- Tenir els discs durs de les màquines partits i un sistema de la gestió d'arrencada (com el *Grub*, per exemple) que permeti triar entre un sistema operatiu i l'altre.

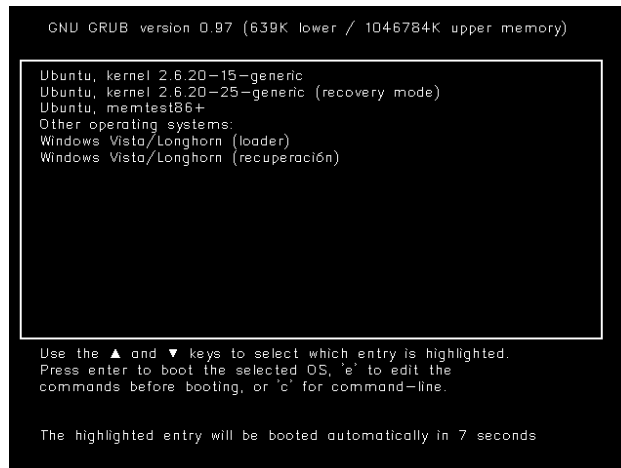


Figura 1: Gestor d'arrencada GRUB

- Aprofitar el sistema operatiu existent (recordem, Windows 7) i utilitzar alguna plataforma de virtualització compatible (VirtualBox, VMware).

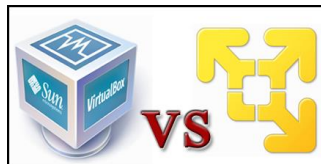


Figura 2: VirtualBox i VMware

Arribat aquest punt, s'havia de prendre la decisió entre una de les dues opcions anteriors, cosa que comprometria absolutament la manera en que es desenvoluparia el projecte. La primera opció presentava una sèrie de problemes de seguretat, explicats a l'apartat corresponent a les decisions preses, que van fer que fos desestimada. Així que vam triar la segona opció que consistia en fer màquines virtuals, instal·lar-hi Linux i el programari de gestió de *cloud* escollit.

Un cop triada la plataforma d'implementació, calia veure si realment aquesta podia arribar a suportar l'execució d'un *cloud*. En altres paraules, calia comprovar si un *cloud* es pot implementar amb màquines virtuals. Com es podrà comprovar més endavant, sí que ho pot fer, així que per aquesta part, el projecte era viable.

Així i tot, encara queden altres aspectes a tenir en compte. En primer lloc, parlarem del maquinari que es necessitava per la realització del treball.

- Un ordinador per buscar la informació necessària, amb accés a Internet. A més també havia de permetre realitzar documentació. Per realitzar aquestes tasques, vaig fer servir un dels ordinadors disponibles al laboratori de BCDS.
- Un ordinador amb suficients recursos com per poder mantenir l'execució de 3 màquines virtuals. En aquest cas, la màquina de què parlava en el punt anterior, no era suficient degut a limitacions de memòria. En conseqüència, vaig utilitzar un servidor del departament que, durant el primer quadrimestre, ens va servir per poder fer les pràctiques de l'assignatura d'Administració de

Sistemes, les quals requerien d'una màquina amb prou potència com per aguantar l'execució de moltes màquines virtuals alhora.

- Una altra màquina que disposés del sistema operatiu Windows 7 i prou recursos com per poder executar 3 màquines virtuals. En aquest cas, podia fer servir qualsevol de l'aula on s'ha implementat el *cloud*, però en aquest cas vaig decidir fer servir el meu ordinador personal, ja que quan vaig necessitar aquest recurs, encara hi havia docència.
- La darrera màquina necessària serviria per poder administrar l'aula. En aquest cas, hem utilitzat la mateixa del primer punt, és a dir, la del laboratori de BCDS.

Seguidament, analitzarem el *software* i les llicències que es necessitaven:

- Una llicència de Windows 7 per la màquina del laboratori, la qual em va proporcionar l'administrador del departament.
- Una llicència pel sistema operatiu del servidor, en aquest cas ESXi 5.5, que era gratuïta.
- La llicència de Windows 7 del meu ordinador personal és la educativa que podem obtenir els estudiants al *Dreamspark* de manera gratuïta, també.
- El sistema operatiu utilitzat pels nodes del *cloud* era l'Ubuntu Server, el qual és lliure i gratuït.
- Com a plataforma de virtualització s'ha utilitzat l'VMware Player, de caràcter gratuït, també.

Finalment, la inversió d'hores per part meva ha estat important, ja que he hagut de consultar moltíssima documentació, aprendre com funciona l'*OpenStack*, dissenyar i provar multitud d'*scripts* tan per Windows com per Linux... Tot i així, com que només cursava dues assignatures a part de realitzar el treball, per aquesta part també era viable.

Les conclusions sobre l'estudi de viabilitat són:

- L'entorn *cloud* i aula docent són compatibles.
- Disposem dels recursos de maquinari.
- També disposem dels recursos de programari.
- Tenim el temps necessari.

Així doncs, el treball final de grau és viable.



### 3. Metodologia

La metodologia de treball emprada per desenvolupar aquest treball s'ha basat en estructurar la feina a fer en una sèrie de tasques principals que seguirien un ordre concret i seqüencial. A partir de l'esquema inicial, aquestes tasques es dividirien en tasques secundàries, per poder tenir una estructura organitzada.

D'altra banda, s'anava informant al tutor periòdicament sobre l'evolució del treball com ara els dissenys fets, les proves realitzades o les decisions que s'anaven prenent; tanmateix, per les decisions que no estaven massa clares, s'han pres conjuntament.

Així, el conjunt de tasques és el següent:

- **Investigació CMPs.** Explorar les diverses opcions de codi lliure que existeixen per crear i administrar *clouds*, conegudes en anglès com CMPs (*Cloud Management Platforms*), per poder decidir-me per una (OpenStack) i fer-la servir com a pal de pallar dels diferents escenaris muntats a posteriori.
- **Escenari de proves ESXi.** Aquesta tasca es corresponia a treballar sobre un escenari semblant al de l'aula. Es composava de diverses subtasques:
  - Aprenentatge. Calia familiaritzar-se amb les eines de virtualització que ofereix l'VMware ESXi.
  - Disseny. Tenint en compte els requeriments de la CMP escollida, dissenyar un entorn amb les eines de treball actuals.
  - Implementació. Instal·lació i configuració de l'escenari bàsic per l'OpenStack.
  - Implementació. Instal·lació i configuració dels serveis de l'OpenStack. Aquesta tasca també comprèn aquestes subtasques:
    - Aprenentatge de les eines.
    - Investigació i comprensió dels nous conceptes.Cal tenir en compte que aquesta part és una de les de més duració del treball, degut a la novetat i a que es realitzaven diverses tasques en paral·lel.
  - Documentació de la feina que s'estava desenvolupant. L'anava realitzant al final de cada dia, així que es pot considerar com una subtasca paral·lela a les anteriors.
  - Proves sobre l'escenari. S'havien de comprovar tots els requisits pel que feia a la part *cloud* de l'aula.
- **Exportació del *cloud* a l'aula.** Degut a l'èxit obtingut a la tasca anterior, ara calia aplicar tot el que s'havia après per poder executar el *cloud* a l'aula. Les subtasques van ser les següents:
  - Investigació. Per aprofitar la feina feta fins ara, vaig haver d'explorar si existia alguna eina que em permetés exportar les màquines virtuals que composaven el *cloud* que havia muntat a l'ESXi. D'altra banda també vaig explorar diverses eines que em permetessin automatitzar les tasques de parada i engegada de les estacions i del *cloud* en sí.
  - Aprenentatge. Familiarització amb les noves eines.

- Disseny. Idear un escenari encara més pròxim que l'anterior (ESXi) en quant a característiques per comprovar que les màquines virtuals seguien fent funcionar el *cloud*.
  - Implementació. Configuració de l'escenari anterior per al seu correcte funcionament.
  - Documentació de la feina.
  - Proves sobre l'escenari. Comprovacions sobre el correcte funcionament del *cloud* sobre l'escenari actual.
- **Implementació del *cloud* a l'aula.** En aquesta etapa es va fer el desplegament de les màquines virtuals que composaven l'OpenStack al llarg de tota l'aula. Les subtasques que la descriuen són:
    - Disseny. Aquesta subtasca va comportar diverses subtasques més, que són:
      - Topologia de la xarxa.
      - Assignació dels diferents tipus de nodes virtuals d'OpenStack a cadascuna de les estacions de l'aula.
    - Implementació. Aquesta part també és formada per vàries subtasques:
      - Configuració prèvia de les estacions per crear l'entorn d'execució del *cloud*.
      - Disseny i creació d'un *script* pels nodes virtuals, per poder ser configurats adientment segons la seva ubicació, és a dir, en quina màquina real s'està executant.
      - Còpia de les màquines virtuals corresponents a les estacions on pertoquessin, aspecte contemplat a la part de disseny de la topologia.
    - Documentació de la feina.
    - Proves sobre l'escenari. Comprovacions sobre el correcte funcionament del *cloud* sobre l'escenari actual.
- **Sistema de gestió i administració.** Aquesta era la darrera etapa/tasca del projecte per assolir els requeriments inicials. Consistia en idear un sistema per poder controlar l'aula en les seves dues vessants, tan com aula docent com *cloud*. Les subtasques eren les següents:
    - Estudi previ de les tasques que havia de realitzar aquest sistema.
    - Aprenentatge. Durant aquesta tasca em vaig haver de documentar sobre totes les eines que necessitaria.
    - Disseny. Decidir concretament i idear com calia que fos el sistema.
    - Implementació. Instal·lació i configuració del sistema de control.
    - Documentació de la feina.
    - Proves. Comprovar que les tasques assignades es desenvolupaven correctament.
- **Proves finals.** En aquesta etapa calia comprovar que el sistema muntat a l'aula es podia arrencar i parar de manera remota, a més de poder programar aquestes a hores que convinguessin.

- **Documentació.** Aquesta darrera tasca es va anar fent al llarg del desenvolupament del projecte, però no pas en el format en què s'està aplicant aquí, sinó com a quadern de bitàcola, per no perdre detall de tota la feina realitzada. La redacció de la memòria final s'ha realitzat un cop acabades totes les tasques anteriors.

## 4. Planificació

La planificació d'aquest treball es va veure influenciada directament per dos factors:

- Metodologia de treball escollida, on s'han definit clarament els passos a seguir per aconseguir els objectius proposats.
- Calendari de docència de l'aula. Degut al meu desig de no afectar la integritat de les màquines de l'aula, vaig decidir la realització d'un escenari de proves, cosa que va afectar tan a la metodologia de treball com a la planificació.

Llavors, per fer la planificació del projecte, només calia seguir els passos establerts a la metodologia de treball, ordenar-los i repartir-los. En el següent diagrama de Gantt es pot observar la planificació amb més detall.

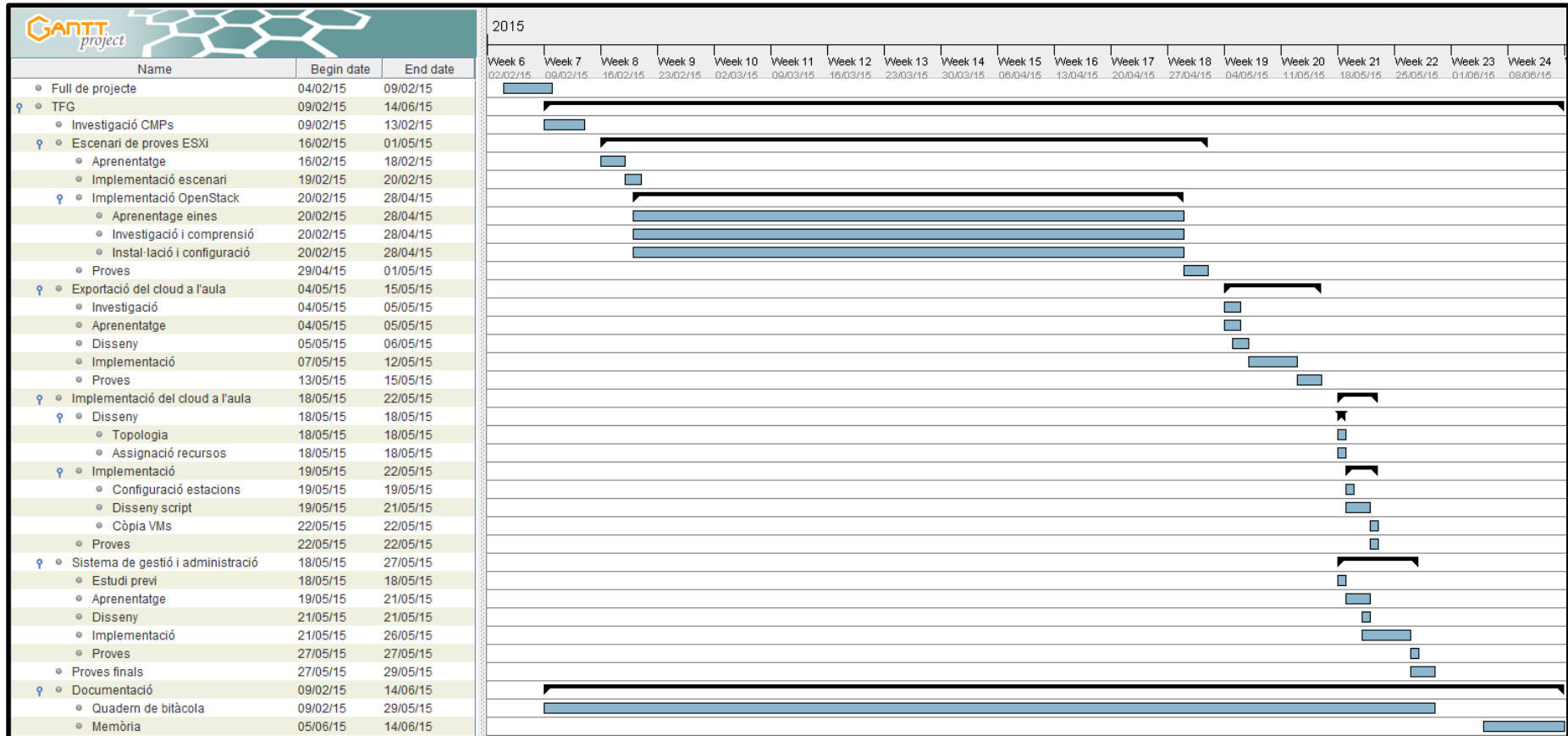


Figura 3: Diagrama de Gantt de la planificació

## 5. Marc de treball i conceptes previs

### 5.1 Cloud Computing

Com s'ha definit a la introducció: “conjunt de serveis accessibles a través de la xarxa (ja sigui Internet o una xarxa privada) que un conjunt de proveïdors ofereixen, ja que disposen d'una sèrie de recursos informàtics compartits (xarxes, servidors, emmagatzemament, aplicacions i serveis) per poder fer-ho. A més, aquests recursos s'han d'oferir als seus consumidors de manera transparent, sense complicacions i sense la intervenció directa del proveïdor”.

Existeixen diversos tipus:

- **Públics.** Són els que disposen de serveis oberts per l'ús del públic, ja siguin gratuïts o de pagament. En aquest tipus de *clouds*, tan les dades com els processos dels clients es barregen als servidors i altres infraestructures, fent això que els usuaris no sàpiguen què està passant exactament al seu servidor. L'accés sempre és remot.

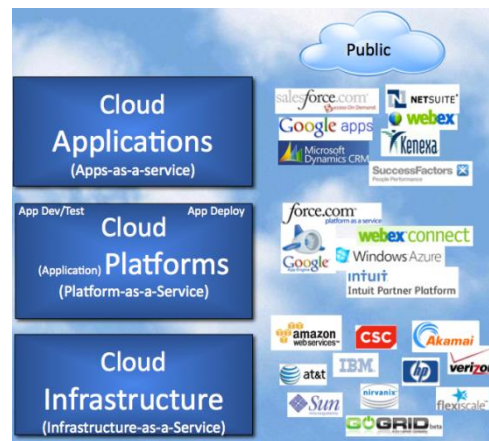


Figura 4: Clouds Públics

- **Privats.** Són operats normalment per una sola organització i gestionats internament per aquesta o per terceres. Requereixen un major grau de compromís però permeten instal·lar-hi el *hardware* desitjat, escollir la gestió de recursos i el control de processos i dades en tot el núvol. És la opció a tenir en compte per grans empreses, ja que és molt més costosa, en canvi, permet més control i és més segura.

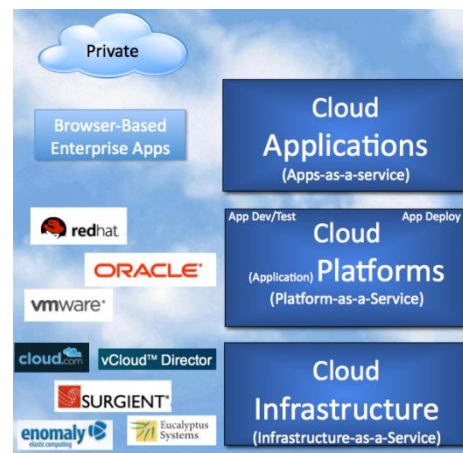


Figura 5: Clouds Privats

- **Híbrids.** Són una composició de dos o més tipus de *clouds* (públic, privat o comunitari), mantenint les entitats separades però lligades entre elles oferint els beneficis dels diferents models. Per exemple, fer servir un *cloud* privat

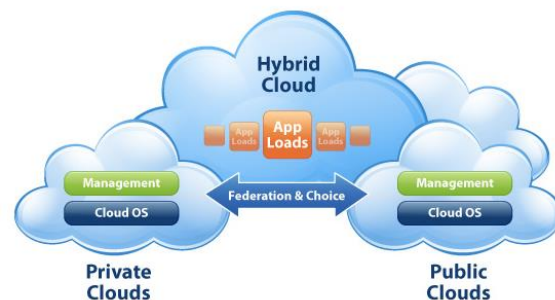


Figura 6: Clouds Híbrids

però per augmentar la capacitat de recursos en algun moment donat on la demanda és més gran, executar certes aplicacions al núvol públic. Aquest procés és anomenat “*Cloud Bursting*”. També es fa servir un núvol públic (treballant amb el privat) per recuperar dades o quan es preveu un tall en el subministrament d’energia.

- **Comunitaris.** Es basen en una infraestructura compartida entre diverses organitzacions amb un propòsit comú. Els costos es reparteixen en un nombre d’usuaris menor que en les públiques, per tant només alguns dels possibles estalvis del núvol s’aconsegueixen.



Figura 7: Clouds Comunitaris

El conjunt de tipus de serveis *cloud* són tres, que són:

- **Infrastructure-as-a-Service (IaaS).** Es tracta d’un mitjà de lliurar emmagatzematge bàsic i capacitats de còmput com a serveis estandarditzats a la xarxa. Servidors, sistemes d’emmagatzematge, connexions, enrutadors, i altres sistemes es concentren, per exemple a través de la tecnologia de la virtualització, per gestionar tipus específics de càrregues de treball depenent de la demanda de l’usuari.



Figura 8: IaaS

- **Platform-as-a-Service (PaaS).** En aquest model, s’ofereix a l’usuari un conjunt d’eines pel desenvolupament d’aplicacions, proves de *software* o l’administració de contingut entre d’altres. D’aquesta manera es poden crear i controlar aplicacions sense haver d’instal·lar cap *hardware* ni controlar la infraestructura.

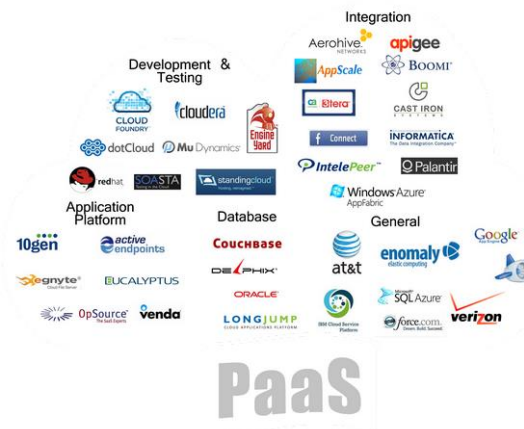


Figura 9: PaaS

- **Software-as-a-Service (SaaS).** Es basa en tenir a disposició de l'usuari un conjunt d'aplicacions que es subministren per la demanda d'aquest. Aquestes aplicacions són accessibles des del navegador o de qualsevol *software* dissenyat per aquest fi, on l'usuari no té cap control sobre la seva configuració (o gairebé cap) però s'estalvia haver d'instal·lar el *software* i evita el cost de suport i manteniment.



## 5.2 Màquina virtual

Conjunt de fitxers que representen l'emulació d'una màquina real amb els seus recursos mitjançant *software* (*hypervisors*).

## 5.3 Hypervisor

Es tracta d'un programari que permet crear i visualitzar màquines virtuals.

## 5.4 Instància

Conjunt de recursos *hardware* d'un *cloud* que conformen una màquina virtual dins aquest entorn. En altres paraules, és una màquina virtual que té una imatge que li permet emular l'execució d'un sistema operatiu, amb les seves aplicacions i serveis.

## 5.5 Imatge

Fitxer que conté l'estat d'un sistema operatiu amb les aplicacions que tingui instal·lades, en una màquina concreta, en el moment que aquest ha sigut creat. D'aquesta manera, si es disposa de les eines necessàries, hom pot prendre aquest fitxer i utilitzar-lo per una màquina compatible, restaurar-lo al disc dur d'aquesta i disposar de l'estat de la màquina original, sense haver de realitzar ni instal·lacions ni configuracions.

## 5.6 API Rest

Conjunt d'operacions que permeten interactuar amb un servidor que les ofereix (servei web), normalment sobre HTTP. Així, el format de les peticions són URLs que segueixen els estàndards GET, POST, PUT i DELETE del propi protocol. Així, si per

exemple fem una petició GET a un servidor a la URL <http://www.myserver.com/items>, aquest retornarà els ítems que tingui.

## 5.7 Cloud Management Platform

Una plataforma d'administració de *clouds* consisteix en un sistema de *software* que configurat convenientment permet gestionar els recursos *hardware* sobre els que estigui instal·lat per oferir-los de manera transparent als seus usuaris.

Per interactuar amb el sistema, existeixen 2 maneres:

- A través d'una API Rest.
- Mitjançant una aplicació web.

Sigui quina sigui la interfície utilitzada per utilitzar els recursos disponibles del *cloud*, aquestes han de permetre que un usuari normal pugui, com a mínim:

- Administrar els recursos que tingui disponibles.
- Disposar d'imatges per la creació de noves instàncies.
- Monitoritzar la utilització dels seus recursos.
- Crear noves instàncies.
- Accés a les instàncies.

En cas de tractar-se d'un usuari administrador, aquest ha de ser capaç de gestionar, entre d'altres:

- Usuaris.
- Imatges.
- Monitoreig dels recursos.
- *Hypervisors* i nodes de computació.
- Instàncies.
- Plantilles d'instàncies.
- Xarxes de les instàncies.

## 5.8 CMPs de codi lliure

Existeixen moltes opcions en quant a plataformes que permeten l'administració de *clouds*, però em vaig centrar en l'estudi dels 3 casos principals, que són:

- OpenStack.
- OpenNebula.
- Eucalyptus.

A continuació explicaré cadascuna de les opcions



## 5.8.1 OpenStack



Figura 11: Logo OpenStack

El projecte OpenStack és una plataforma de *cloud computing* basada en codi lliure que ofereix suport per *clouds* públics o privats. El que persegueix aquest projecte és una implementació simple, escalabilitat massiva (horitzontal) i un conjunt molt complet d'eines; pel qual colaboren experts en *cloud computing* d'arreu del món.

Es proveeix una pila de *cloud* a nivell d'infraestructura, és a dir, *Infrastructure-as-a-Service*, mitjançant la interacció d'un nombrós conjunt de serveis que més endavant explicaré. Cada servei ofereix la seva API perquè els seus usuaris la puguin utilitzar i així poder automatitzar tasques fàcilment.

### Característiques principals

- Escalable horitzontalment. Es poden afegir tants nodes de computació com es desitgi, podent-los organitzar en zones de disponibilitat (*availability zones*).
- Virtualitzable. Tots els nodes que conformen un *cloud* OpenStack poden ser virtualitzats, és a dir, màquines virtuals. De cara a un administrador o a un usuari final, aquest fet és transparent.
- Compatible amb l'API de l'AWS.
- Servei de dades basat en *block storage* o bé *object storage*.
- Autoservei. Disposa d'una aplicació web per als seus usuaris que els permet gestionar els seus recursos, ja siguin màquines virtuals, emmagatzematge o dispositius de xarxa.
- Administració. De la mateixa manera que els usuaris normals disposen de l'aplicació web, l'administrador del *cloud* disposa d'un perfil dins aquesta aplicació web que els permet administrar el conjunt de recursos del *cloud*, administrar les zones de disponibilitat, controlar els nodes de computació (migracions, recuperació de desastres), generar noves plantilles de màquines virtuals, noves imatges...

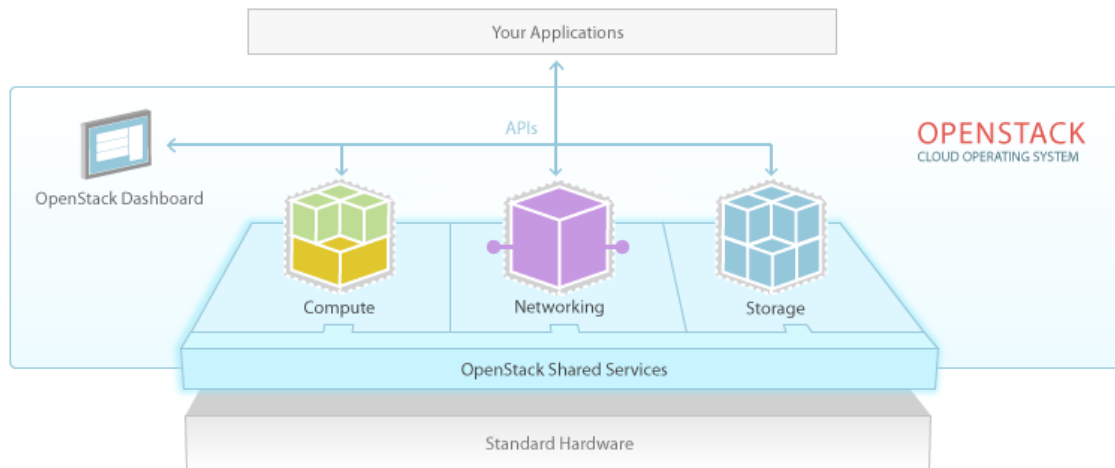


Figura 12: Serveis OpenStack

### Eines disponibles

Les eines de que disposa l'OpenStack són els seus serveis en sí, els quals estan explicats en el seu apartat corresponent, concretament el 5.8.

### Requisits del sistema

- L'arquitectura mínima d'un entorn OpenStack es basa en 3 nodes (si es tria el mode de xarxa *neutron*). Aquest fet fa que es necessitin un mínim de 3 màquines. D'altra banda, gràcies a que és virtualitzable, aquestes màquines poden ser virtuals i, per tant, executar-se sobre la mateixa màquina real, sempre i quan aquesta compleixi amb els requisits mínims de *hardware*. Durant la implementació del *cloud*, he arribat a poder executar-lo en una màquina *quad core @2.66Ghz* amb tan sols 8GB de RAM, sobre un Windows 7.
- Els requisits mínims pel node *Controller* són: 1 processador, 2GB de RAM i 5GB d'emmagatzematge.
- Pel *Network*: 1 processador, 512MB de RAM i també 5GB d'emmagatzematge.
- Pel *Compute*: 1 processador, 2GB de RAM i 10GB d'emmagatzematge.
- Els sistemes operatius suportats són openSuse 13.2, SUSE Linux Enterprise Server 12, Red Hat Enterprise Linux 7, CentOS 7, Fedora 21 i Ubuntu 14.04 LTS.
- Pel que fa a la xarxa, el *Controller* necessita 2 interfícies, el *Network* 3, i el *Compute* 2. Totes elles poden treballar a 100Mb.
- El *software* necessari és l'NTP, una base de dades (*MariaDB*, per exemple) i un servei de missatgeria MQ (*RabbitMQ*, per exemple)

### 5.8.2 OpenNebula



Figura 13: Logo OpenNebula

L'OpenNebula és una plataforma d'administració de *clouds*, molt simple d'instal·lar i configurar però amb moltíssimes eines, permetent virtualitzar *datacenters* per fer-los servir com a *cloud* públic, privat o híbrid, a nivell d'infraestructura. Està enfocat com a eina per les empreses.

### Característiques principals

- Autoservei. Catàlegs d'imatges, xarxes i plantilles d'VMs. Control i monitoreig dels recursos virtuals.
- Virtualitzable. Tots els nodes que conformen un *cloud* OpenNebula poden ser virtualitzats, és a dir, màquines virtuals. Per un administrador o un usuari final, aquest fet és transparent.
- Suport pel control i monitoreig d'aplicacions multicapa elàstiques, o serveis que necessiten diverses VMs interconnectades; tot mitjançant les regles d'*auto-scaling*.
- Administració i monitoreig dels recursos del cloud, alta disponibilitat, generació de clústers, multizona...
- Servei de dades basat en sistemes de fitxers distribuïts (NFS, Lustre, GlusterFS, ZFS...), *datastores* de l'VMware, *datastores* LVM i Ceph.

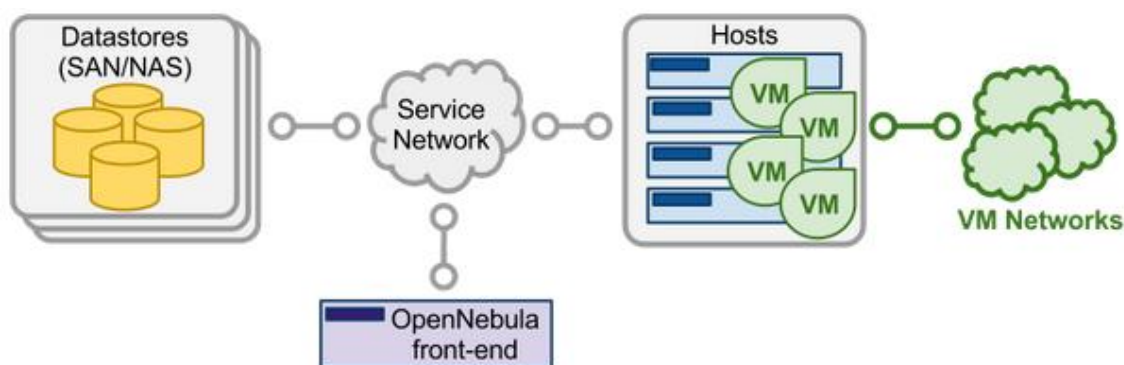


Figura 14: Serveis OpenNebula

### Requisits del sistema

- L'arquitectura bàsica es conforma per un node anomenat *Front-end* que executa els serveis de l'OpenNebula, nodes anomenats *Hosts* que són els que tenen el KVM i proveeixen els recursos *hardware*, centres de dades on s'emmagatzemen les imatges per les VMs i xarxes físiques que es fan servir per connectar les operacions de control de l'OpenNebula amb els servidors d'emmagatzematge i VLANs per les màquines virtuals.
- Els requisits *hardware* són, per executar el *software* en conjunt, 512MB de RAM, 3GB de disc i un nucli de 64 bits pel sistema operatiu. De fet, des de la web d'OpenNebula es pot descarregar el disc dur d'una màquina feta amb VirtualBox que, seguint les configuracions indicades al manual, només arrencar-la ja es pot accedir a través de la seva aplicació web a l'entorn d'administració i ús d'un *cloud*.
- Els sistemes operatius suportats són el Red Hat Enterprise Linux (6.5 i 7.0), l'Ubuntu Server 14.04 LTS, CentOS (6.5 i 7.0), Debian 7.1, VMware (ESX 5.5 i vCenter 5.5), XEN (3.2 i 4.2).
- L'*hypervisor* és el KVM.

### 5.8.3 Eucalyptus

L'Eucalyptus és un *software* lliure que serveix per crear *clouds* privats o híbrids que són compatibles amb l'AWS d'Amazon. Està definit a nivell d'*Infrastructure-as-a-*



Figura 15: Logo Eucalyptus

Service, permetent administrar i proveir els recursos *hardware* (computacionals i d'emmagatzemament) segons la demanda. També ofereix característiques de *high-availability* (alta disponibilitat), fent del *cloud* que es construeixi un entorn robust.

### Característiques principals

Les seves característiques principals són:

- **Compatible amb l'API de l'AWS**, per permetre fer servir eines que siguin familiars pels usuaris dels AWS.
- **Configuracions d'alta disponibilitat**, per garantir la robustesa del *cloud* i que sigui tolerant a errors.
- **Servei de dades** basat en blocs (*block storage*) o bé en cubs (*bucket storage*), com els EBS o els S3 d'Amazon.
- **Autoservei**. S'ofereix una consola pels usuaris del *cloud*, perquè aquests puguin sol·licitar els recursos que necessitin, sense la intervenció d'un administrador.
- **Consola gràfica per l'administrador**. El *Dashboard* és una interfície gràfica que permet als administradors del *cloud* configurar, administrar i monitoritzar el *cloud*.

### Eines disponibles

A més, disposa d'una sèrie d'eines perquè els seus usuaris puguin administrar els seus recursos, que són:

- **Administració de claus SSH**, que serveixen per validar la identitat dels usuaris per fer *login* a les seves VMs.
- **Administració d'imatges**: permet empaquetar, pujar, enregistrar, descriure, baixar, desempaquetar i treure del registre les imatges de les màquines virtuals.
- **Support tan per Linux com per Windows** com a sistemes operatius de les màquines virtuals.
- **Administració de les adreces IP**, amb el que es pot reservar, associar, disgregar, descriure i alliberar adreces IP. Depenent del mode en que es faci treballar la xarxa, es pot tenir accés a IPs públiques que no estan associades a VMs estàticament (IPs elàstiques). Es permet als usuaris que reservin i associïn dinàmicament aquestes IPs elàstiques a les seves VMs.
- **Administració de grups de seguretat**, amb el que es permet crear regles de tallafocs que s'associen a les VMs que formen part del grup pel qual s'estiguin definint. Aquests grups es poden crear, descriure, eliminar, autoritzar i revocar.
- **Administració de volums i *snapshots* (estat concret d'una VM)**, cosa que permet crear volums dinàmics basats en blocs, de la mateixa manera que també es permet crear, vincular a una VM, desvincular-los, empaquetar-los o eliminar-los. També es poden crear o eliminar *snapshots* i fins i tot crear nous volums a partir d'*snapshots*.

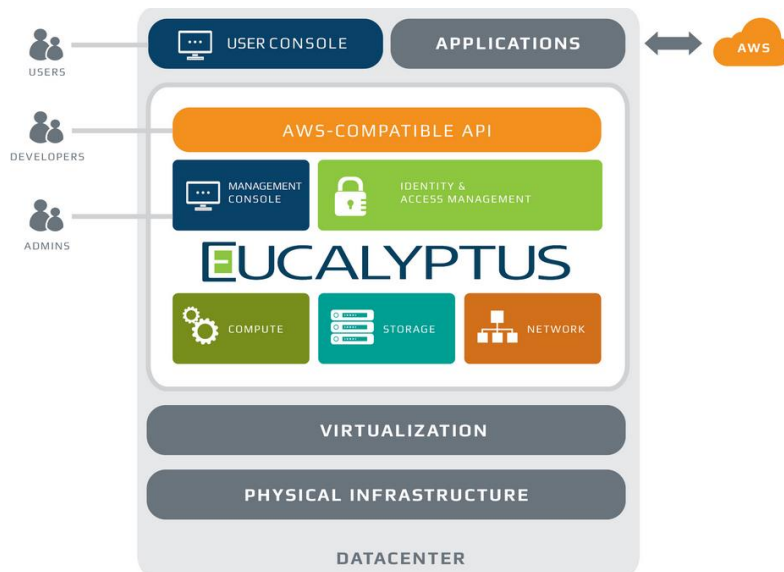


Figura 16: Serveis Eucalyptus

## Requisits del sistema

### Computacionals

- Màquines físiques. Tots els components de l'Eucalyptus s'han d'instal·lar a màquines físiques, no pas a màquines virtuals.
- CPUs. Es recomana que cada màquina disposi de com a mínim d'una CPU Intel o AMD de 2 cores.
- Sistema operatiu. S'ha d'instal·lar sobre una de les següents distribucions Linux: CentOS 6 o RHEL 6, tots dos només la versió de 64 bits.
- Es necessita un mecanisme de sincronització temporal, com ara l'NTP.
- *Hypervisors*: els sistemes operatius anteriors han de tenir instal·lat i configurat el KVM.
- Accés remot: tots els nodes han de tenir l'SSH configurat i accés *root* via *sudo*.

### Emmagatzematge i memòria

- Cada màquina de la xarxa necessita un mínim de 30GB d'emmagatzematge.
- Es recomanen com a mínim 100GB pels nodes que continguin VMs Linux i un mínim de 250GB pels que continguin VMs Windows. Val a dir que quant més espai disponible hi hagi, més màquines virtuals es podran crear.
- Cada màquina de la xarxa necessita un mínim de 4GB de RAM.

### Xarxa

- Les interfícies de xarxa han de ser com a mínim 1Gb Ethernet.
- Com a mínim tots els nodes han de disposar d'una interfície de xarxa, tot i que per un funcionament mínimament òptim i qüestions d'aïllament de xarxes, seria convenient un mínim de 2.
- Algunes configuracions requereixen que el node que s'encarrega de controlar el clúster disposi d'un mínim de 2 interfícies d'1Gb Ethernet.
- Depenent de la configuració també, es podria donar el cas de la necessitat de disposar de VLAN *trunking*.
- Disposar de 2 rangs d'IPs dedicats.

## 5.9 Wake On Lan (WOL)

És un estàndard del protocol Ethernet que permet arrencar ordinadors que estan apagats mitjançant l'enviament d'uns paquets anomenats "*màgic packet*". Els requisits són:

- La màquina que s'ha d'engegar ha de treballar sobre Ethernet.
- La seva interfície de xarxa ha de permetre el WOL i tenir-lo activat.
- Disposar de l'adreça MAC de la màquina que es desitja arrencar.
- Normalment les interfícies de xarxa *wireless* no ho permeten.

## 5.10 Deepfreeze

Es tracta d'un programari que serveix per congelar/descongelar les màquines, entre d'altres utilitats. Congelar una màquina significa que quan un administrador es decideix a congelar la màquina, aquesta romandrà en l'estat en què es troba en el moment de la congelació fins que es decideixi descongelar per realitzar algun canvi.

Aquesta eina és de gran utilitat dins el món TIC, sobretot dins entorns Windows, ja que permet als administradors disposar de les estacions en estats estables, sense que per causa d'usuaris o virus, aquestes es desconfigurin o malmetin. A més, si es disposa del *software* d'administració (*Deepfreeze Console*), aquest permet controlar les estacions que tingui associades, poden arrencar-les per la xarxa, parar-les, reiniciar-les, accés remot, control d'actualitzacions, execució de processos remots...

## 5.11 VMware ESXi

Sistema operatiu dedicat a la creació, administració i execució de màquines virtuals que utilitzen els recursos *hardware* de la màquina on estigui instal·lat.

## 5.12 VMware Player

Programari que permet crear, administrar i executar màquines virtuals que utilitzen els recursos *hardware* de la màquina on estigui instal·lat.

## 5.14 VMware VIX

Conjunt d'eines per l'VMware Player/Workstation que permet realitzar a través de la línia de comandes les mateixes operacions que es farien amb les màquines virtuals a través de la interfície gràfica.

## 6. Requisits del sistema

### 6.1 Requisits OpenStack

Un *cloud* OpenStack (versió Juno) que utilitzi el mode de xarxa *neutron* necessita d'un mínim de 3 nodes per funcionar. Les característiques *hardware* mínimes per cadascun d'ells és:

- Pel node *Controller* cal una màquina amb 1 processador, 2GB de RAM i 5GB d'emmagatzematge, 1 targeta de xarxa.
- Pel node *Network*: 1 processador, 512MB de RAM i també 5GB d'emmagatzematge, 3 targetes de xarxa.
- Pel node *Compute*: 1 processador, 2GB de RAM i 10GB d'emmagatzematge, 2 targetes de xarxa.

Tot i això, la bibliografia directament recomana les següents especificacions *hardware*:

- Pel node *Controller* cal una màquina amb 1/2 processadors, 8GB de RAM i 100GB d'emmagatzematge, 1 targeta de xarxa.
- Pel node *Network*: 1/2 processadors, 2GB de RAM, 50GB d'emmagatzematge, 3 targetes de xarxa.
- Pel node *Compute*: 2/4/>4 processadors, >8GB de RAM i >100GB d'emmagatzematge, 2 targetes de xarxa.

La topologia de xarxa que s'espera configurar pels nodes de l'OpenStack és la següent:

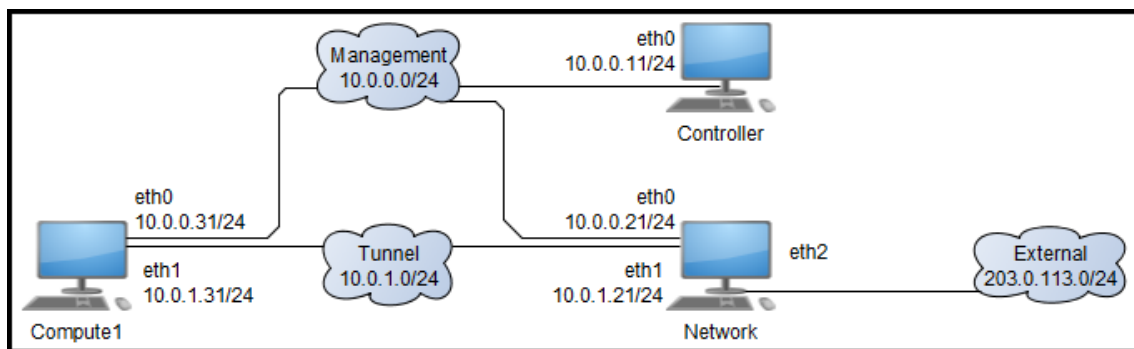


Figura 17: Topologia bàsica de xarxa OpenStack

Totes les màquines s'han de poder arrencar des de la xarxa.

D'altra banda, també hi ha requisits *software*, que són:

- En primer lloc, s'ha de decidir entre una de les distribucions Linux que suporta l'OpenStack, les quals estan enumerades a l'apartat 5.8.1, requisits del sistema, d'aquest mateix document.
- En segon lloc cal tenir instal·lats i configurats els següents serveis:
  - NTP, per sincronitzar els nodes de l'infraestructura.
  - Servidor mySQL, perquè els serveis OpenStack tinguin on guardar i recuperar informació.

- Servei de missatgeria MQ, perquè els serveis OpenStack es puguin enviar missatges entre ells.

Finalment, cal disposar de l'estructura de xarxa física necessària per poder assolir els requisits dels nodes.

Tenint en compte que tot el sistema pot implementar-se utilitzant l'eina de la virtualització, el cert és que si es disposa d'una màquina que disposi de la suma dels requisits dels 3 nodes, es pot muntar el *cloud* directament en aquesta única màquina.

Un altre avantatge a considerar de la virtualització és el relatiu a la xarxa, ja que les màquines virtuals tindran interfícies de xarxa també virtuals i, per tant, la màquina real només cal que en disposi d'una.

## 6.2 Requisits servidor d'aula

La màquina que tindrà el rol de servidor de l'aula, ha de complir amb els següents requisits *hardware*:

- Processador de com a mínim 2 cores, pot ser de 32 o 64 bits, amb un mínim de 2.5Ghz.
- Memòria de 3GB o més.
- Una interfície de xarxa que treballi com a mínim a 100Mbps.
- 50GB d'espai a disc.

Els requisits *software* són:

- Sistema operatiu Windows Server 2003.
- Tenir instal·lada la consola del Deepfreeze.
- Algun mecanisme que permeti enviar paquets de *Wake On Lan*.

L'accés remot d'aquest servidor ha d'estar habilitat, per poder complir amb els requeriments del treball.

## 6.3 Requisits ordinador de treball

Per poder treballar amb els nodes d'OpenStack i el servidor d'aula, només cal una màquina que disposi d'algun tipus d'interfície de xarxa que es pugui connectar a Internet i a la xarxa on estiguin connectades les màquines que s'hagin d'instal·lar i configurar.



## 7. Estudis i decisions

### 7.1 Plataforma d'administració del *cloud*

La primera decisió que calia prendre era triar una de les plataformes d'administració de *clouds* (CMPs) explorades. Per poder decidir entre una o altra, calia tenir en compte les característiques de l'entorn on s'acabaria instal·lant, l'aula III-04i.

Tal i com s'explica a l'apartat 5.8, dins dels requisits per poder instal·lar qualsevol de les 3 plataformes, els seus nodes han de tenir com a sistema operatiu una de les distribucions Linux dels seus respectius llistats de distribucions compatibles.

D'altra banda, a l'aula d'informàtica, els sistemes operatius de les estacions de treball són *Windows 7 Enterprise*, cosa que plantejava, com s'ha explicat a l'estudi de viabilitat, dues possibilitats:

- Partir els discos, realitzar una instal·lació d'un Linux i d'un sistema gestor de l'arracada (*Grub*, per exemple). Aquesta opció es va desestimar degut a que des del Linux, amb permisos de *root* es pot fer amb la màquina el que es vulgui. Això vol dir que no només es podria fer malbé el Linux i l'OpenStack, sinó que també es podria fer malbé el Windows i d'aquesta manera no estar complint amb un dels requisits bàsics del treball, que és la independència de les funcions de l'aula, com a docent i com a *cloud*.
- Treballar amb màquines virtuals. A les estacions de l'aula hi ha instal·lades les dues plataformes de virtualització gratuïtes disponibles, l'VMware Player i el VirtualBox. Aquesta opció passaria per virtualitzar els nodes del *cloud* mitjançant qualsevol de les dues plataformes que s'acaben d'esmentar.

Per tant, veient els perills que suposava implementar una arrencada dual a l'aula, es va triar virtualitzar el *cloud*. Per tant, la CMP que s'escollís ho havia de permetre, i no és el cas d'Eucalyptus, tal i com diu als seus requisits del sistema, així que va ser descartat en aquest punt.

En canvi, les altres dues plataformes sí que es podien virtualitzar, així que calia analitzar altres aspectes per decidir entre una o altra:

- Pel que fa als **requisits de hardware**, cap de les dues suposava un problema, donat que les especificacions de les estacions de l'aula compleixen de sobres.
- Si ens fixem en les funcionalitats i característiques, pràcticament disposen del mateix.
- Les dues plataformes són modulars, es basen en la interacció d'una sèrie de mòduls o serveis que en conjunt, proporcionen un sistema d'administració de recursos *hardware*.

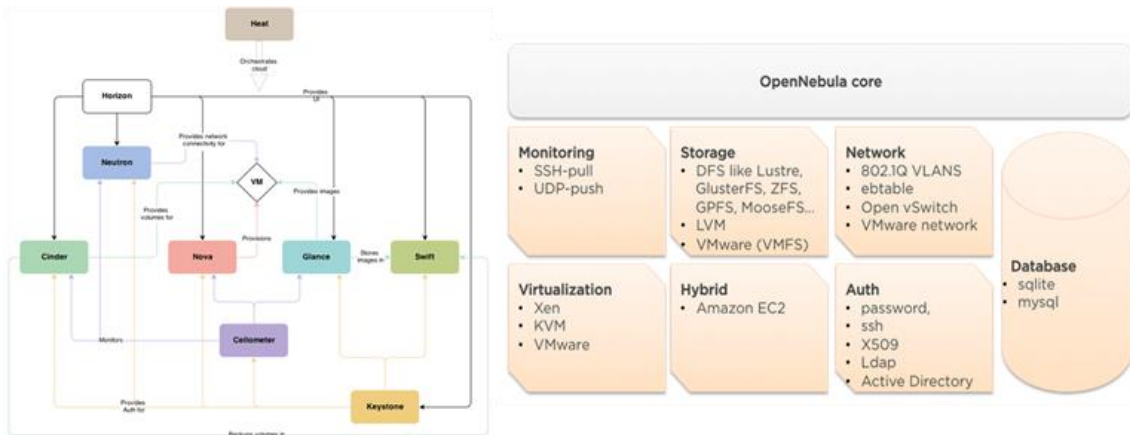


Figura 18: Comparativa serveis OpenStack i OpenNebula

- Totes dues disposen d'una àmplia documentació per administradors, constructors, usuaris.
- La diferència que existeix entre les dues és que l'OpenNebula natiu es pot instal·lar d'una manera ràpida, fàcil i transparent, sense haver de configurar gairebé res. En canvi amb l'OpenStack natiu cal instal·lar i configurar cadascun dels seus serveis. Tot i això, existeixen versions com la Ubuntu OpenStack, la RDO (Red Hat) o la Mirantis (CentOS/Ubuntu), que permeten implementar un *cloud* OpenStack sense haver de configurar els serveis *core*.
- Totes dues plataformes són fetes servir per múltiples empreses:
  - OpenNebula: Akamai, BBC, Fermilab, Terradue, SURFsara, Prohuban, Netways, Esa, Cloudweavers, China Mobile, CentOS, Blackberry, tuntastic, Bit.nl, Avalon.hr, Rentalia, Cenatic, Csuc, Deloitte, Onlindata, Unity...
  - OpenStack: CiscoWebEx, PayPal, Wikimedia Foundation, Cardiff School of Computer Science & Informatics, Florida State University, Harvard University, MIT Computer Science & Artificial Intelligence Lab, NEC, Spanish National Research Council, University of Zurich, Asus, Seagate, Telefónica, TRUSTe, Intel, Lenovo, American Express, BBVA, Yahoo...
- Les dues plataformes estan en constant evolució.

Així doncs, sobre quina plataforma decidir? Doncs bé, finalment em vaig decidir per OpenStack, per dos motius:

- Els usuaris d'aquesta plataforma, segons la informació que OpenStack proporciona a la seva web, són més nombrosos i tenen més relleu internacional, a més d'aparèixer multitud d'universitats, com és el nostre cas.
- El fet de poder triar sobre muntar una versió nativa o bé una amb una capa de software.

Un altre motiu, no menys important, és que fins a l'any 2013, l'OpenStack és la CMP lliure de la que més se'n parla i més participants té, tal i com se'n parla en aquest article<sup>1</sup>. A més és la que ha experimentat un creixement més elevat des del seu origen:

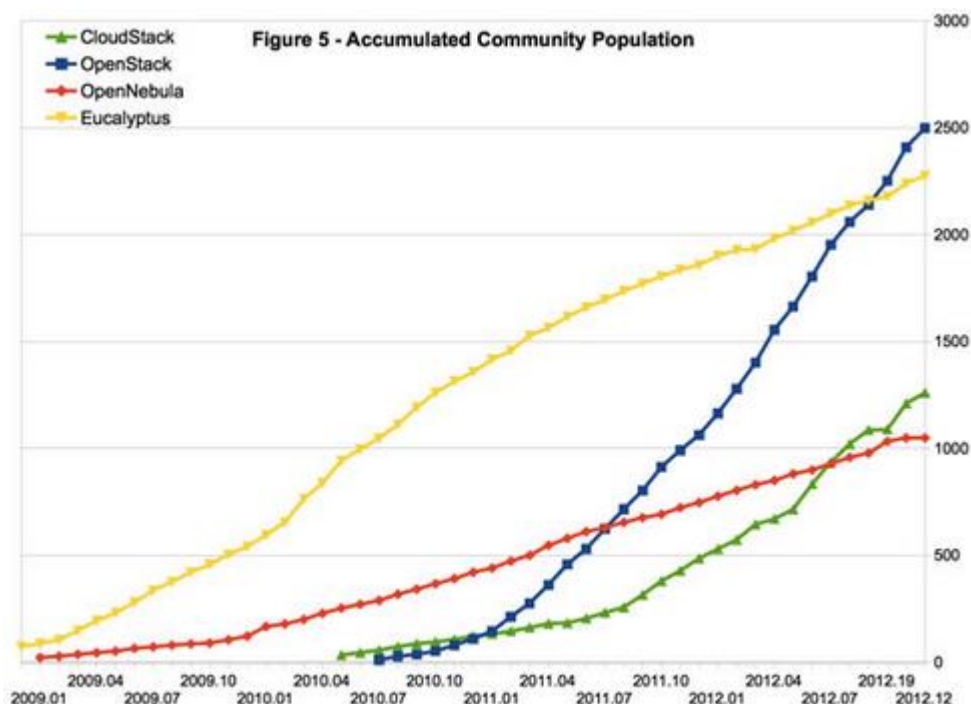


Figura 19: comparativa creixements de les diferents CMPs

## 7.2 Escenaris

Segons el que ja s'ha explicat a l'apartat de la metodologia de treball (concretament l'apartat 3), he dissenyat, implementat i provat 3 escenaris, requerint, cadascun d'ells, l'ús de diferents eines que a continuació explicaré segons l'escenari.

### 7.2.1 ESXi

#### Maquinari

Per poder muntar aquest escenari, vaig utilitzar un servidor amb les següents característiques:

- Processador de 4 nuclis @3.1Ghz.
- 32GB de memòria RAM.
- 2 discs d'1TB cadascun.
- 2 interfícies de xarxa d'1Mbps.

Per accedir al servidor, buscar informació i redactar el quadern de bitàcola ho feia des d'un dels ordinadors del laboratori de BCDS, que tenia les següents característiques:

<sup>1</sup> <http://www.revistacloudcomputing.com/2013/01/comparativa-de-las-plataformas-cloud-abiertas-openstack-opennebula-eucalyptus-y-cloudstack/>

- Processador de 2 nuclis @2.66Ghz.
- 4GB de memòria RAM.
- 1 disc de 160GB.
- 1 interfície de xarxa de 100Mbps.

### Programari

El *software* que es va utilitzar és el següent:

- Com a sistema operatiu pel servidor, l'*VMware ESXi v5.5*.
- Com a sistema operatiu de les màquines virtuals que representen l'OpenStack, l'Ubuntu Server 14.04 LTS (64 bits), simplement perquè és la distribució Linux que prefereixo.
- Serveis i programari relacionat amb l'OpenStack. Aquest conjunt té el seu apartat específic degut a la seva extensió.
- Com a sistema operatiu de la màquina virtual que s'encarregava de gestionar la interconnectivitat de les xarxes virtuals, el *pfSense 2.1.5*.
- El sistema operatiu de la màquina d'accés (la del laboratori de BCDS) era un Windows 7 Enterprise (64 bits).
- L'accés a la consola d'administració i d'*hypervisors* de les màquines virtuals, anomenada *VMware vSphere*, es feia mitjançant un client *vSphere*.

## 7.2.2 Exportació del *cloud* a l'aula

### Maquinari

En aquest cas, vaig utilitzar el meu ordinador personal, que té les següents característiques:

- Processador de 4 nuclis @3.2Ghz.
- 8GB de memòria RAM.
- 1 disc SSD de 256GB.
- 1 interfície de xarxa d'1Mbps.

### Programari

El programari que vaig utilitzar en aquesta etapa, va ser:

- El sistema operatiu del meu ordinador personal és un Windows 7 Professional (64 bits), el més semblant al que hi ha a l'aula.
- Per extreure les màquines del servidor ESXi vaig utilitzar una eina anomenada *VMware Converter*, que és la única que hi ha per realitzar aquesta tasca.
- Com a *hypervisor* vaig triar l'*VMware Player*, tot i poder utilitzar el *VirtualBox*, per diversos motius:
  1. Es tracta de la plataforma que es fa servir normalment a les TIC, tal i com ens van fer veure a les dues visites realitzades durant el curs de l'assignatura d'Administració de Sistemes (Josep Blanes, Salvador Salanova).
  2. L'eina de virtualització que he fet servir més durant la carrera ha estat el *VirtualBox*, i per tant, no estaria aprenent gairebé res nou en aquest aspecte, exceptuant l'automatització de tasques de les VMs a través de CLI (*Command Line Interface*).

3. He utilitzat la plataforma *ESXi d'VMware* en la primera part del treball.

Tot i haver triat aquesta opció, no hi ha cap problema per migrar d'una plataforma a una altra, ja que totes dues tenen suport per a l'*Open Virtualization Format*, un format de màquina virtual que fa de pont entre totes dues. Per poder fer la conversió, només cal seguir els passos descrits aquí<sup>2</sup>.

- Degut a haver triat l'*VMware Player*, també calia familiaritzar-se amb dues eines més; d'una banda les *VMware VIX*, que servien per utilitzar les màquines virtuals a través de línia de comandes i, d'altra banda, amb el *Virtual Network Editor*, que permet administrar les xarxes virtuals que es desitgi tenir a l'*hypervisor*.

Amb aquestes noves eines, vaig poder implementar, arrencar i provar el *cloud* al meu ordinador personal, pas previ i molt pròxim a l'escenari final, l'aula.

### 7.2.3 Escenari final

#### Maquinari

Per l'escenari final, el *hardware* que s'ha utilitzat és en primer lloc les 18 màquines de l'aula, que tenen les següents característiques:

- Processadors de 4 nuclis però 8 *threads* disponibles @3.4Ghz.
- 12GB de memòria RAM.
- Discs durs de 500GB.
- 5 interfícies de xarxa d'1Mbps.

Per controlar-les, vam decidir utilitzar la màquina que es feia servir per accedir al servidor *ESXi* durant el primer escenari, descrita a l'apartat (7.2.1). Per accedir a totes les màquines anteriors (inclòs el servidor que les administra) de manera remota, he utilitzat el meu portàtil, que té les següents especificacions:

- Processador de 2 nuclis amb 4 *threads* @2.6Ghz.
- 4GB de memòria RAM.
- 1 disc SSD de 256GB.
- 1 interfície de xarxa *wireless*.
- 1 interfície de xarxa d'1Mbps.

Les màquines de l'aula estan connectades a un mateix *switch* a 100 Mbps, amb una topologia de xarxa que s'explicarà a l'apartat d'implementació.

Com s'ha comentat anteriorment, es va fer servir la màquina que tenia al laboratori de BCDS com a servidor d'administració de les estacions de l'aula. Això no és ben bé així, en realitat el que vaig fer va ser deixar el sistema operatiu que ja tenia instal·lat intacte i fer una màquina virtual que s'encarregués de l'administració de l'aula. D'aquesta manera, el servidor no depèn del *hardware* on s'estigui executant, disposant així d'un servidor portable i independent de la plataforma. El sistema operatiu triat per la màquina virtual va ser el Windows Server 2003 Standard Edition, degut a:

---

<sup>2</sup> <http://www.howtogeek.com/125640/how-to-convert-virtual-machines-between-virtualbox-and-vmware/>

- El *software* d'administració de que disposem pel *Deepfreeze* és per Windows.
- El Windows disposa de les *Pstools*, que permetran executar *batch files* en remot, sense haver de tenir en compte incompatibilitats.
- Dins dels Windows Servers, aquest consumeix pocs recursos
- Només ha d'administrar una aula de 18 estacions.
- Pels serveis que ha d'oferir, no cal que tingui un sistema operatiu que ofereixi moltíssimes eines.
- L'accés que s'ha de fer a aquesta màquina és des de la xarxa local de BCDS.

### Programari

El programari utilitzat en aquesta darrera etapa va ser el mateix que a la segona, tret que:

- El Windows de les màquines és la versió Enterprise, però de 64 bits també.
- El sistema operatiu de la màquina que controla les estacions de l'aula és un Windows 2003 Server.
- Tots els scripts que s'han escrit per automatitzar les tasques de Windows han estat escrits com a *batch files*.
- També s'ha automatitzat la configuració de cadascun dels nodes OpenStack de l'aula, com a *scripts* de Linux.
- Per poder executar programes i comandes remotament s'han utilitzat les *PsTools*, ja que són compatibles amb els *batch files*.
- Per administrar la congelació, descongelació i més opcions de les estacions, s'ha fet servir la consola del *Deepfreeze* a la màquina que les administra.
- El *software* que ha permès realitzar l'arrencada des de la xarxa ha sigut el *WakeOnLan*, d'*Aquila Technology*, ja que permet utilitzar-se des de la línia de comandes, i en conseqüència, des de *batch files*.
- Per tenir accés remot al servidor de l'aula i poder administrar-lo sense haver de fer modificacions a cap element de xarxa (*routers*, *switchs*) de la universitat, he utilitzat l'eina *Teamviewer*, que permet tenir un escriptori remot a través d'Internet gràcies a la seva aplicació.

El funcionament, configuració i resultats de totes les eines *software* descrites fins ara, es podrà entendre a l'apartat d'implementació.

## 7.3 Serveis de l'OpenStack

No es pot entendre el funcionament de l'OpenStack sense entendre què és el que fan cadascun dels seus serveis, i per tant, a continuació explicaré cadascun dels que he implementat. Cal tenir en compte però, que no s'han muntat tots els serveis

disponibles, ja que els requeriments del treball no els demanen i hem decidit conjuntament amb el tutor de no muntar-los. Els serveis que no s'han muntat són:

- *OpenStack Block Storage (cinder)*, *OpenStack Object Storage (swift)*, ja que són serveis que es dediquen a l'emmagatzematge, i aquest *cloud* només ha d'actuar com a centre de càlcul, no pas d'emmagatzematge.
- *Database service (trove)*, que s'encarrega de proporcionar bases de dades com a servei, és a dir, instàncies que treballen únicament com a bases de dades (relacionals o no), que pel nostre cas d'ús tampoc calia.

Malgrat que no es van instal·lar, si en un futur es desitja afegir aquestes funcionalitats al *cloud*, sempre es pot fer, sense cap problema. Seguidament, parlaré dels que sí que vaig instal·lar i configurar.

### 7.3.1 Identity service (Keystone)

#### Funcions del servei

- Seguiment dels usuaris i els seus permisos.
- Provisió d'un catàleg de serveis disponibles amb les seves APIs corresponents.

Quan s'instal·la aquest servei, s'han d'enregistrar tots els serveis involucrats en l'arquitectura OpenStack que s'hagi muntat. Així, l'**Identity** pot portar un seguiment dels serveis OpenStack que hi ha instal·lats i quina és la seva ubicació dins la xarxa.

#### Definicions per entendre l'Identity

- **User** (Usuari). Representació digital d'una persona, sistema o servei que utilitza els serveis cloud de l'OpenStack. L'**Identity** valida que les peticions d'entrada són fetes realment per l'usuari que les està fent. Els usuaris tenen unes credencials per fer *login* i se'ls han d'anar assignant *tokens* que els permetin tenir accés als recursos. Els usuaris també poden ser assignats a un *tenant* (contenedor) o projecte i comportar-se com si estiguessin continguts en aquest.
- **Credentials** (Credencials). Són les dades que confirmen la identitat d'un usuari. Exemples: usuari i contrasenya; usuari i clau per l'API, o un *token* d'autenticació lliurat pel servei **Identity**.
- **Authentication** (Autenticació). És el procés de confirmació de la identificació d'un usuari. El servei **Identity** confirma una petició d'entrada validant una sèrie de credencials que l'usuari ha proporcionat. Aquestes credencials en un principi són l'usuari i la contrasenya, o un usuari i una clau per l'API. Quan es validen les credencials, l'**Identity** emet un *token* d'autenticació que l'usuari ha de fer servir per les següents peticions.
- **Token** (Tiquet). És una cadena alfanumèrica que es fa servir per accedir a les APIs de l'OpenStack i els seus recursos. Un *token* pot ser revocat en qualsevol moment i és vàlid durant un període de temps concret.
- **Tenant** (contenedor, projecte). És un contenidor que es fa servir per agrupar o aïllar una sèrie de recursos. També serveixen per agrupar o aïllar objectes d'identitat. Depenent del servei on es defineixi, un *tenant* pot ser un client, un compte d'usuari, una organització o un **projecte**. La definició que en fa la

Wikipedia és prou clara també: “*És un grup d’usuaris que comparteixen la mateixa vista del software que aquests fan servir.*”

- **Multitenancy.** És un principi d’arquitectura de *software* on una sola instància del *software* que s’executa en un servidor serveix diversos *tenants*. En aquest paradigma, les aplicacions es dissenyen per proveir a cada *tenant* un tros compartit i dedicat de la instància *software*, incloent les seves dades, configuració, administració d’usuaris, funcionalitats específiques per cada *tenant* i propietats no funcionals. En contrast a la filosofia *Multitenancy* existeix la *Multi-instance*, on es creen diverses instàncies per servir diferents *tenants*.
- **Service** (servei). Són serveis de l’OpenStack, com ara el **Compute (nova)**, **Object Storage (swift)** o l’**Image Service (glance)**. S’encarreguen de proveir un o més punts d’entrada (*endpoints*) on els usuaris poden accedir als recursos i realitzar operacions.
- **Endpoint** (punt d’entrada). Es tracta d’una adreça de xarxa que permet accedir a un servei; normalment es tracta d’una URL. Si es fan servir extensions per *templates* (patrons), es poden crear *templates* pels *endpoints* per així agrupar els serveis de consum disponibles sota un *template* al llarg de tota la regió. O en altres paraules, tothom accedeix als recursos d’una manera estàndard.
- **Role** (rol). Conjunt de permisos i drets que s’assignen per a realitzar un conjunt d’operacions. En el servei **Identity**, un *token* que s’hagi donat a un usuari inclou una llista de rols. Els serveis que es facin servir per aquell usuari determinen com s’interpreten el conjunt de rols que té l’usuari i quines operacions o recursos pot donar accés cadascun dels rols.
- **Keystone client.** És una interfície de comandes per l’API del servei **Identity**. Per exemple, els usuaris poden executar les comandes “**keystone service-create**” i “**keystone end-point-create**” per registrar serveis a les seves instal·lacions d’OpenStack.



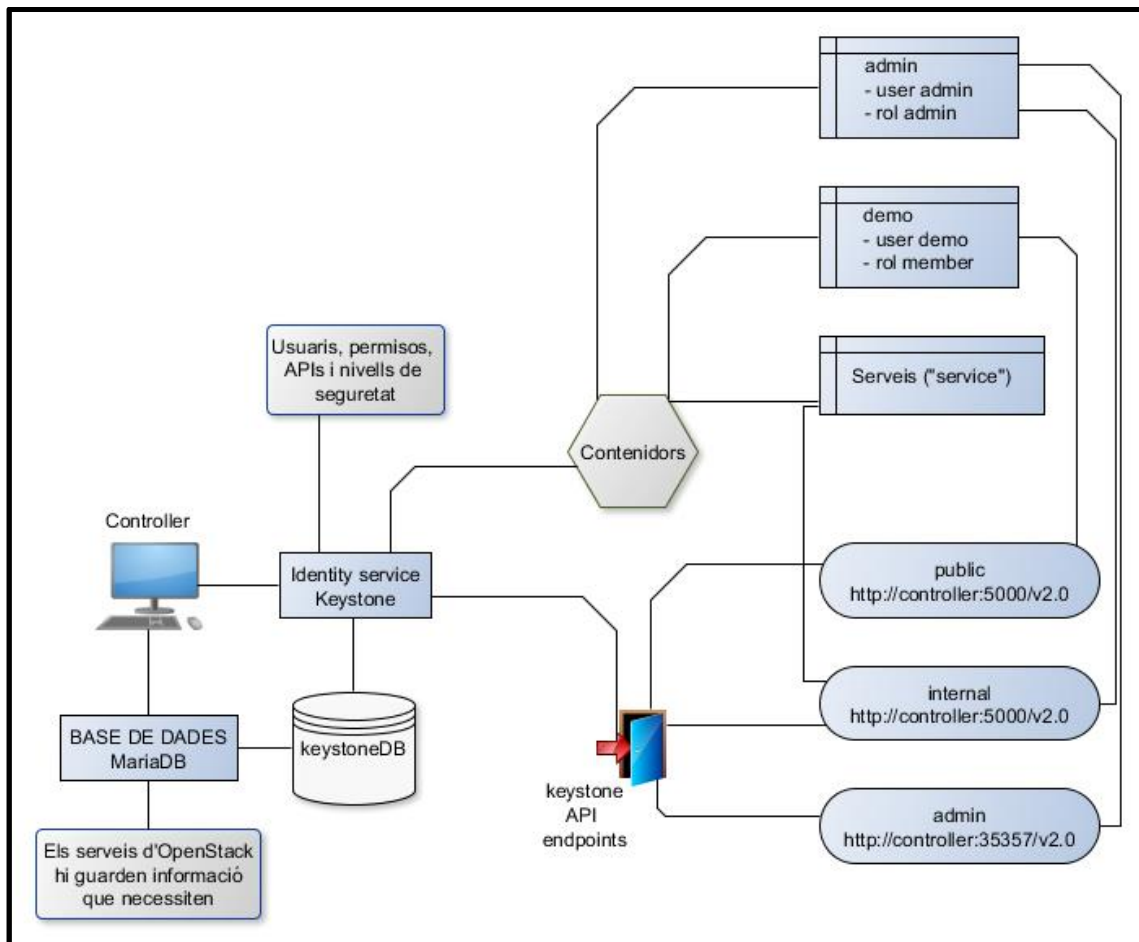


Figura 20: Components Serveis Keystone

### 7.3.2 Image service (Glance)

El servei **Image** proporciona als seus usuaris descobrir, enregistrar i recuperar imatges de màquines virtuals. Ofereix una API Rest que permet sol·licitar metadades de màquines virtuals i recuperar-ne la imatge, també. Les imatges es poden guardar a diferents ubicacions, des de simples sistemes de fitxers fins a sistemes d'emmagatzematge d'objectes com ara l'**Object Storage** de l'OpenStack. Si s'utilitza la versió de sistema de fitxers, les dades es guarden al directori **/var/lib/glance/images/** del node **Controller**.

El **Glance** disposa de diversos processos amb les següents funcions respecte les dades:

- *Caching*.
- Consistència i disponibilitat.
- Auditors, actualitzadors, "reapers".

#### Components del servei

- **glance-api**. S'encarrega d'acceptar crides a l'API per al descobriment, recuperació i emmagatzematge de les imatges.
- **glance-registry**. Emmagatzema, processa i recupera metadades (mida, tipus...) de les imatges. El registre és privat pel servei **Glance**, no ha de quedar exposat als usuaris.

- **Database.** Emmagatzema les metadades de les imatges. Es pot triar la base de dades segons preferència, però la majoria d'implementacions fan servir MySQL o SQLite.
- **Repositori d'imatges.** Hi ha diversos tipus de repositoris suportats, com ara:
  - Sistemes de fitxers normals
  - Emmagatzematge d'objectes
  - Dispositius de blocs RADOS
  - HTTP
  - S3 d'Amazon

Alguns d'aquests sistemes són només de lectura.

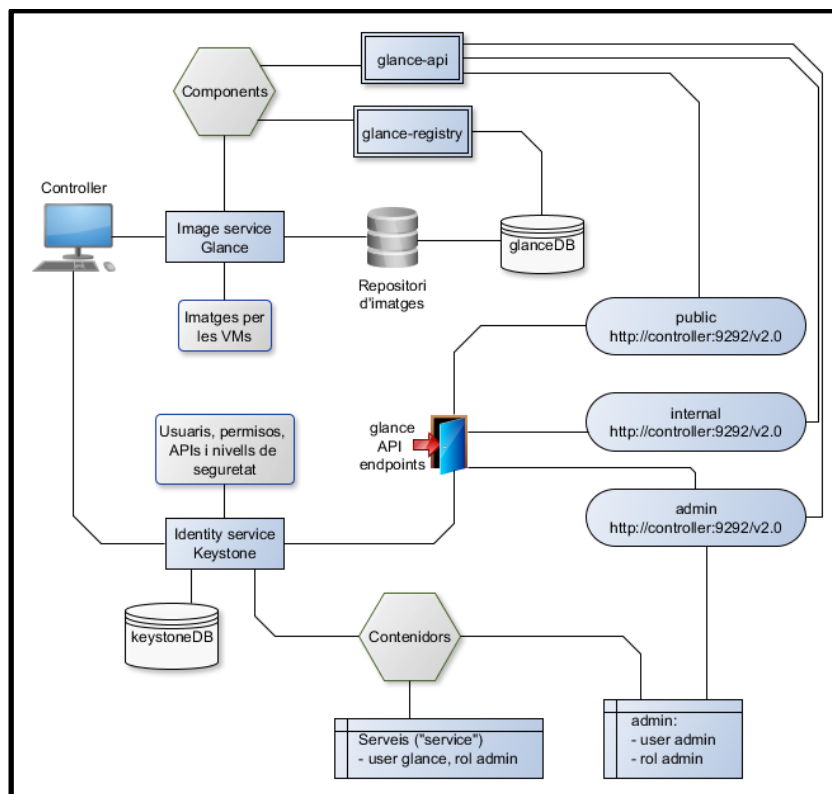


Figura 21: Components servei Glance

### 7.3.3 Compute service (Nova)

Gestiona el cicle de vida de les instàncies de computació: creació, programació (*scheduling*) i eliminació de màquines virtuals sota demanda. Interactua amb l'**Identity (keystone)** per l'autenticació, amb l'**Image (glance)** per les imatges de disc i servidors i amb el **Dashboard (horizon)** per la interfície d'usuari i la d'administració.

**Àrees del servei:**

**API:**

- **nova-api service.** Servei que atén crides a l'API de computació. Dóna suport a l'*OpenStack Compute API*, l'*EC2 d'Amazon* i l'*Admin API* per usuaris amb privilegis administratius. Fa complir algunes polítiques i inicia la majoria

d'activitats d'“orquestració” (govern, sincronització, organització), com ara executar una instància.

- **nova-api-metadata service.** Accepta peticions de metadades fetes per les instàncies. Aquest servei generalment s'utilitza quan s'executa en mode *multi-host* amb instal·lacions fetes amb *nova-network*.

#### **Core (nucli) de computació:**

- **nova-compute service.** És un dimoni que crea i elimina instàncies de màquines virtuals a través de les APIs dels *hypervisors*. Per exemple:
  - XenAPI per XenServer/XCP
  - Libvirt per KVM o QEMU
  - VMwareAPI per VMware

Processar és bastant complex. Bàsicament, el dimoni accepta accions de la cua i executa una sèrie de comandes al sistema com ara executar una instància KVM i actualitzant el seu estat a la base de dades.

- **nova-scheduler service.** Agafa una petició d'una instància d'una màquina virtual de la cua i determina a quin node de computació s'ha d'executar.
- **nova-conductor module.** És la interfície entre el servei **nova-compute** i la base de dades, és a dir, el *nova-compute* no té accés directe a la base de dades.

#### **Xarxa per les VMs:**

- **nova-network worker daemon.** Semblant al **nova-compute**, accepta tasques de xarxa de la cua i també manipula la xarxa. Exemple: configurar interfícies que facin de pont o bé canviar regles de les taules IP.

#### **Interfície a través de la consola**

- **nova-consoleauth daemon.** Autoritza els *tokens* subministrats pels *console proxies* (tipus de sistema de màquines virtuals que s'encarrega de presentar una consola a través d'una interfície web) als usuaris. Aquest servei ha d'estar funcionant perquè les consoles via web funcionin, podent atendre'n diferents tipus.
- **nova-novncproxy daemon.** Proveeix un *proxy* (consola web) per accedir a instàncies en execució a través d'una connexió VNC. Suporta clients a través de navegadors web que no siguin VNC.
- **nova-spicehtml5proxy daemon.** Com l'anterior, proveeix una consola web però a través d'una connexió SPICE. Suporta clients a través de navegadors web HTML5.
- **novaxvpnvncproxy daemon.** Igual que el *nova-novncproxy*, però aquest dóna suport per clients Java específics d'OpenStack.
- **nova-cert daemon.** Certificats x509 (claus públiques).

#### **Administració d'imatges (escenari EC2)**

- **nova-objectstore daemon.** Es tracta d'una interfície que dóna suport a instal·lacions que han de poder treballar amb les Euca2ools (eines de línia de comandes per treballar amb l'AWS – *Amazon Web Services*). Així, les eines

euca2ools parlen al *nova-objectstore* i aquest parla amb l'**Image Service (glance)** en el seu llenguatge.

- **euca2ools client**. Eines de la línia de comandes que serveixen per interactuar amb l'AWS.

#### ***Clients des de línia de comandes i altres interfícies***

- **nova client**. Permet als usuaris enviar comandes com a un contenidor administrador o com a un usuari final.

#### ***Altres components***

- **La cua**. Es tracta d'un *hub* central que permet intercanviar missatges entre els diferents dimonis anteriors. Normalment està implementat amb *RabbitMQ*, però també es podria fer amb sistemes AMQP com ara l'*Apache Qpid* o el *Zero MQ*.
- **Base de dades SQL**. Emmagatzema estats relatius a les instàncies (en temps de construcció d'aquestes o d'execució), incloent:
  - Tipus d'instàncies disponibles.
  - Instàncies que s'estan fent servir.
  - Xarxes disponibles.
  - Projectes (contenidors, *tenants*)

Teòricament, el **Compute (nova)** dona suport per qualsevol base de dades que l'*SQL-Alchemy* suporti. Les més comuns són **SQLite3**, **MySQL** i **PostgreSQL**.

Així, l'esquema de components del servei és el següent:

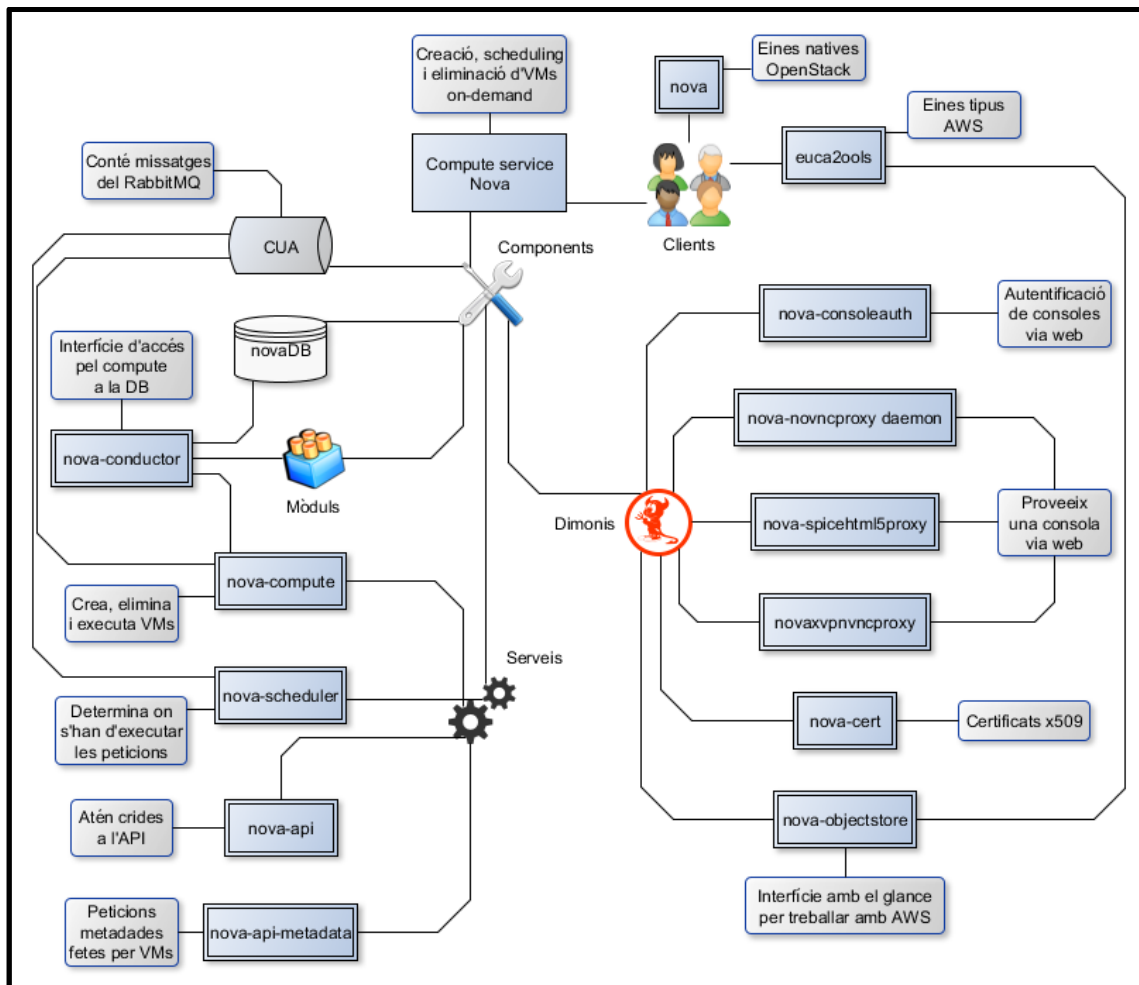


Figura 22: Components servei Nova

### 7.3.4 OpenStack networking (Neutron)

El **neutron** permet habilitar diversos tipus de xarxes per instància a més d'incloure *plugins* per una gran varietat de productes que donen suport per xarxes virtuals. També permet crear i afegir interfícies de xarxa gestionades per altres serveis de l'OpenStack a xarxes. Els *plugins* es poden implementar per adaptar diferents equips de xarxa i *software*, proveint així flexibilitat i desenvolupament a l'arquitectura OpenStack.

#### Components

##### Servidor

- **neutron-server**. Accepta i enruta peticions API cap al *plugin* de xarxa de l'OpenStack adient per servir-les.

##### Plugins i agents de xarxa de l'Openstack

Agrega i treu ports, crea xarxes o subxarxes i proveeix adreçament IP. Aquests *plugins* i agents es diferencien depenent del proveïdor i de les tecnologies que es facin servir al cloud. L'OpenStack Networking (**neutron**) pot treballar amb *plugins* i agents per

*switchs* físics o virtuals Cisco, productes NEC OpenFlow, OpenvSwitch, Linux *bridging*, Ryu Network Operating System i també amb productes VMware NSX.

Els agents més comuns són capa 3, DHCP, i un agent *plugin*.

### **Cua de missatges**

S'ha fet servir a la majoria d'instal·lacions de **neutron** per encaminar informació entre el *neutron-server* i diversos agents, com ara bases de dades per guardar estats de xarxa per *plugins* particulars.

El **neutron (OpenStack Networking)** bàsicament interactua amb el **compute (nova)** per proveir xarxes i connectivitat per les seves instàncies.

### **Conceptes**

El **neutron** s'encarrega de tots els aspectes de xarxa per a la *Virtual Networking Infrastructure* (VNI) i dels aspectes de la capa d'accés per la *Physical Networking Infrastructure* (PNI) dins l'entorn OpenStack. A més permet als contenidors (*tenants*) crear topologies avançades de xarxes virtuals incloent serveis com ara *firewalls*, balancejadors de càrrega i VPNs.

El **neutron** s'encarrega de proveir l'abstracció per les xarxes, subxarxes i *routers*. Cada abstracció realitza les tasques que realitzaria com si fos real: les xarxes contenen subxarxes i els routers encaminen el tràfic entre diferents subxarxes i xarxes.

Qualsevol configuració del **neutron** té com a mínim una xarxa *external*. A diferència de la resta de xarxes, aquesta no és una simple xarxa virtual, sinó que representa una vista de la xarxa externa accessible des de fora de l'OpenStack. És a dir, les IPs d'aquesta xarxa, són visibles per qualsevol de la xarxa externa. Evidentment, el DHCP està inhabilitat en aquesta xarxa.

D'altra banda, qualsevol configuració del **neutron** també ha de disposar d'una o més xarxes internes. Aquestes xarxes *software* connecten directament amb les VMs. Només es pot accedir a aquestes VMs estant a la mateixa xarxa o a una subxarxa. Per tenir accés des de l'exterior (o de l'interior –VMs- cap a fora) d'aquestes xarxes internes, s'ha de fer a través de *routers*.

Adicionalment, es poden mapejar IPs de xarxes externes a ports de la xarxa interna. Quan algun component està connectat a una subxarxa, aquesta connexió s'anomena port. Es poden associar IPs externes amb ports cap a VMs. D'aquesta manera, les entitats externes poden tenir accés a les VMs. Vindria a ser una NAT.

El **neutron** també dóna suport a grups de seguretat (*security groups*), els quals permeten als administradors definir regles per grups al *firewall*. Una VM pot pertànyer a un o més grups de seguretat, i el **neutron** aplica les regles en aquests grups de seguretat per bloquejar o desbloquejar ports, rangs de ports o tipus de tràfic per VMs.

Cada *plugin* del **neutron** fa servir els seus propis conceptes. Encara que no són vitals per fer operar amb l'VNI (*Virtual Networking Infrastructure*) i l'entorn OpenStack, entendre aquests conceptes pot ajudar a configurar el **neutron**. Totes les instal·lacions del **neutron** fan servir un *plugin core* i un grup de seguretat pel *plugin* (o simplement el grup de seguretat per *plugins* "No-Op"). Adicionalment, els *plugins* Firewall-as-a-Service (FWaaS) i el Load-Balancer-as-a-Service (LBaaS) estan disponibles.

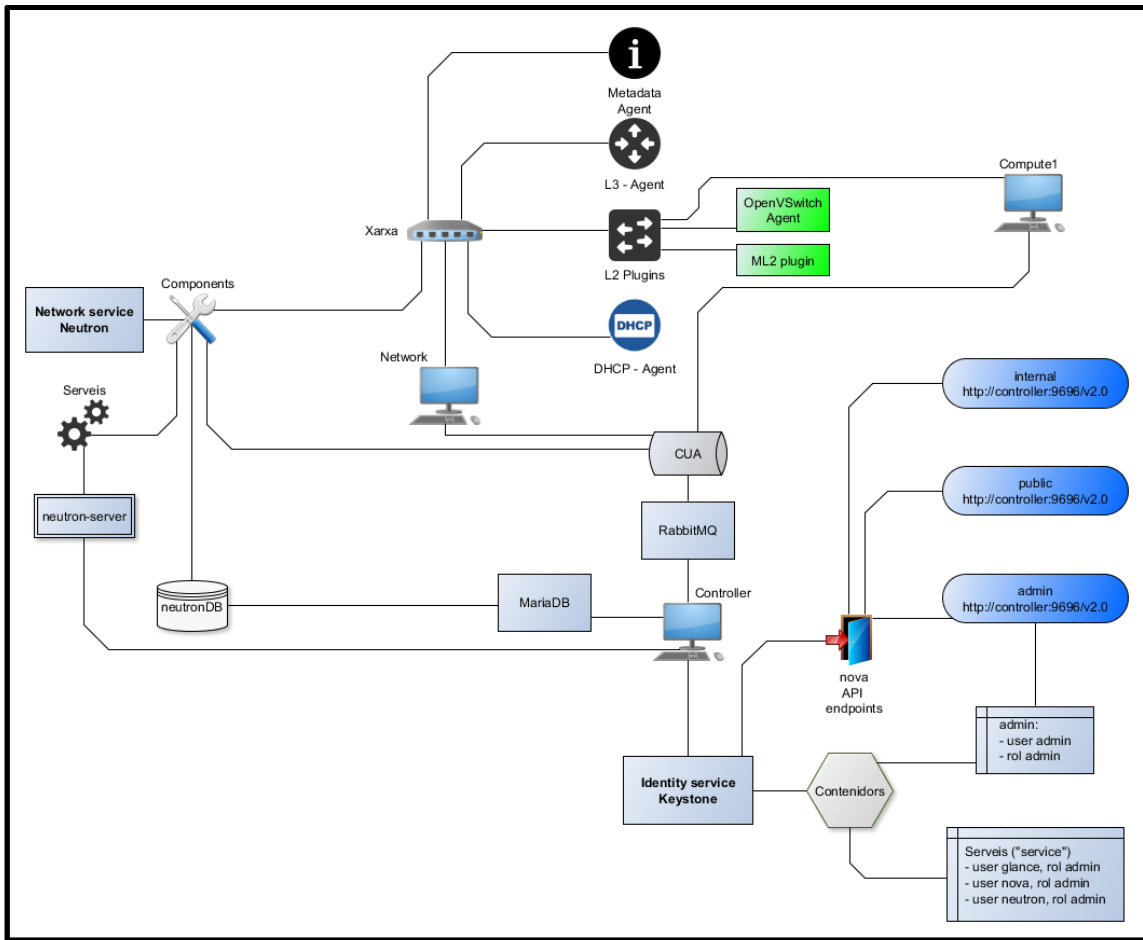


Figura 23: Components servei Neutron.

### 7.3.5 Dashboard (Horizon)

Es tracta d'una interfície web que permet als administradors i als usuaris del cloud gestionar diversos recursos i serveis de l'OpenStack. Habilita interaccions basades en web que utilitzen les APIs que té el **Controller**.

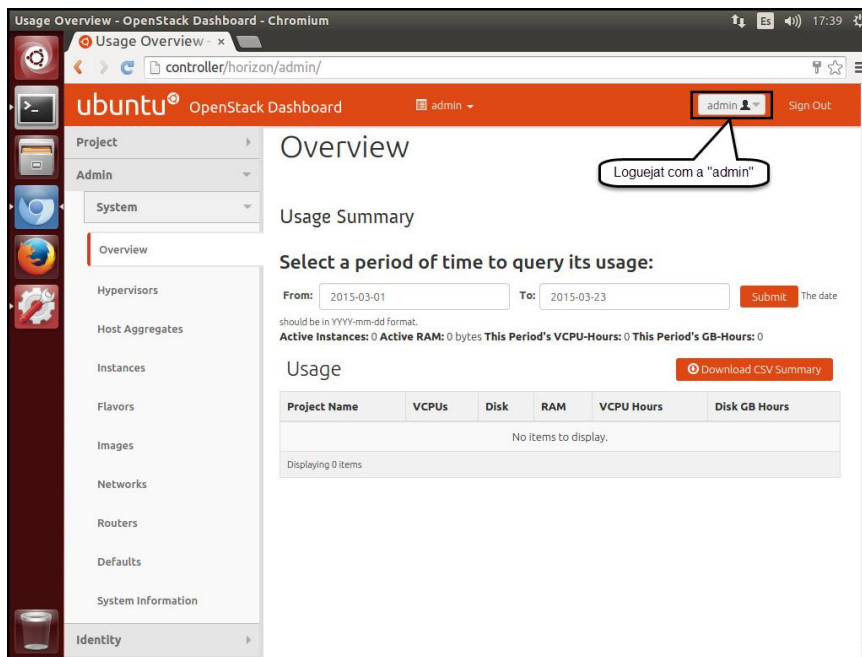


Figura 24: Vista del Dashboard d'un administrador

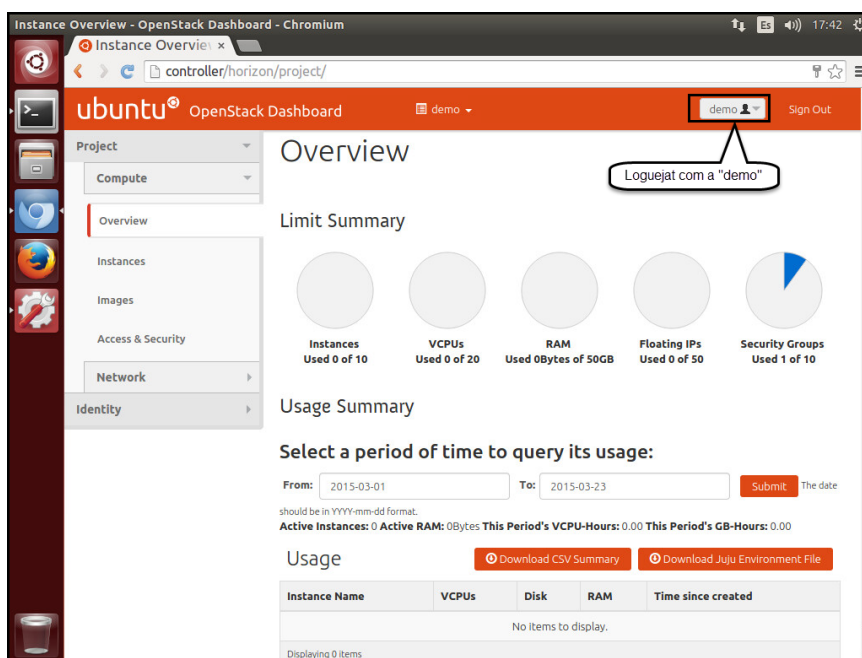


Figura 25: Vista del Dashboard d'un usuari

### 7.3.6 Orchestration module (Heat)

Aquest mòdul el que proporciona als usuaris del cloud és poder crear plantilles (*templates*) de recursos. Amb l'ús d'aquests es poden definir arquitectures que es facin servir freqüentment, automatitzant d'aquesta manera la creació d'aquestes. Dins aquests patrons s'hi poden definir instàncies, *floating IPs*, volums, *security groups* i usuaris. També s'hi poden definir altres paràmetres com ara la *high-avalability* (en el cas d'un servidor, seria la recuperació instantània davant una fallada, transparent a l'usuari) de les instàncies, *auto-scaling* o bé *stacks* aniuades.



Es pot treballar directament amb el **Heat** o bé a través de *plugins*. Els seus components són:

- **Heat command-line client**. Una línia de comandes que es comunica amb l'API del Heat per executar APIs de l'AWS CloudFormation. També es pot fer servir l'API pròpia del Heat (HOT).
- **Heat-api component**. És l'API REST nativa de l'OpenStack pel Heat. Funciona enviant les peticions al **heat-engine** a través d'RPC (Remote Procedure Call).
- **Heat-api-cfn component**. Una API de *queries* de l'AWS que és compatible amb l'AWS CloudFormation. Funciona igual que l'anterior.
- **Heat-engine**. S'encarrega d'executar els *templates* i proveeix esdeveniments (respostes) a l'usuari de l'API.

### 7.3.7 Telemetry module (Ceilometer)

S'encarrega de diverses tasques, entre elles:

- Facilita dades de mesures preses als serveis de l'OpenStack.
- Capta esdeveniments i mesures monitorejant notificacions enviades des dels serveis.
- Publica les dades que recol·lecta a diversos destins, incloent magatzems de dades i cues de missatges.
- Crea alarmes quan troba *data-breaks* (entesos per les normes que s'hagin definit).

Els components que l'integren són:

- **ceilometer-agent-compute** (agent de computació). S'executa en cadascun dels nodes de computació i recol·lecta estadístiques d'ús de recursos.
- **ceilometer-agent-central** (agent central). S'executa en un servidor central administratiu per recol·lectar estadístiques de recursos que no estiguin lligats a les instàncies o als nodes de computació. Es poden executar diversos agents per fer escalar el servei horitzontalment.
- **ceilometer-agent-notification** (agent de notificacions). També s'executa en un (o més d'un) servidor central administratiu, i consumeix missatges de la cua/es per construir dades d'esdeveniments i de mesura.
- **ceilometer-collector** (col·lector). S'executa en un (o més d'un) servidor central administratiu i distribueix les dades de telemetria recollides cap a *data-stores* o bé consumidors externs, sense modificar-les.
- **ceilometer-alarm-evaluator** (avaluador d'alarmes). S'executa en un (o més d'un) servidor central administratiu per determinar quan s'han de disparar

alarmes segons si la tendència de les estadístiques està creuant el límit configurat durant una finestra de temps determinada.

- **ceilometer-alarm-notifier** (notificador d'alarmes). S'executa en un (o més d'un) servidor central administratiu per permetre configurar alarmes basades en l'avaluació de límits per determinades col·leccions de mostres.
- **ceilometer-api** (API server). S'executa en un (o més d'un) servidor central administratiu per oferir accés a les dades del magatzem de dades (*data store*).

### 7.3.8 Data processing service (DPS - Sahara)

El DPS té com objectiu oferir clústers de processament de dades (*Hadoop, Spark*) a usuaris amb pocs medis, simplement especificant diversos paràmetres com ara la versió de *Hadoop*, la topologia del clúster, els detalls del *hardware* dels nodes, entre d'altres. Després de que l'usuari faciliti aquests paràmetres, el **Sahara** proveeix medis per escalar clústers prèviament configurats, afegint i traient nodes sota demanda.

Aquesta nova funcionalitat atén a diversos casos d'ús, com per exemple:

- Proveïment ràpid de clústers Hadoop a l'OpenStack per desenvolupament i QA.
- Utilització de recursos de computació desaprofitats del cloud OpenStack.
- *Analytics-as-a-Service* per ad-hoc o bé càrregues de treball analítiques per ràfegues.

Les característiques clau són:

- Dissenyat com un component de l'OpenStack.
- Es pot administrar via **API Rest** o bé des del **Dashboard**.
- Dóna suport per diferents distribucions del Hadoop:
  - El sistema està preparat per poder acoblar-li conjunts Hadoop (*Hadoop installation engines*).
  - Integració amb eines d'administració específiques com ara *Apache Ambari* o bé *Clouder Management Console*.
- Plantilles per configuracions de Hadoop, amb la possibilitat de modificar-ne diversos paràmetres.
- Interfície d'usuari amb bona usabilitat per peticions d'anàlisis ad-hoc basada en *Hive* o *Pig*.

Si posem tots els serveis que he explicat junts, l'esquema final és:

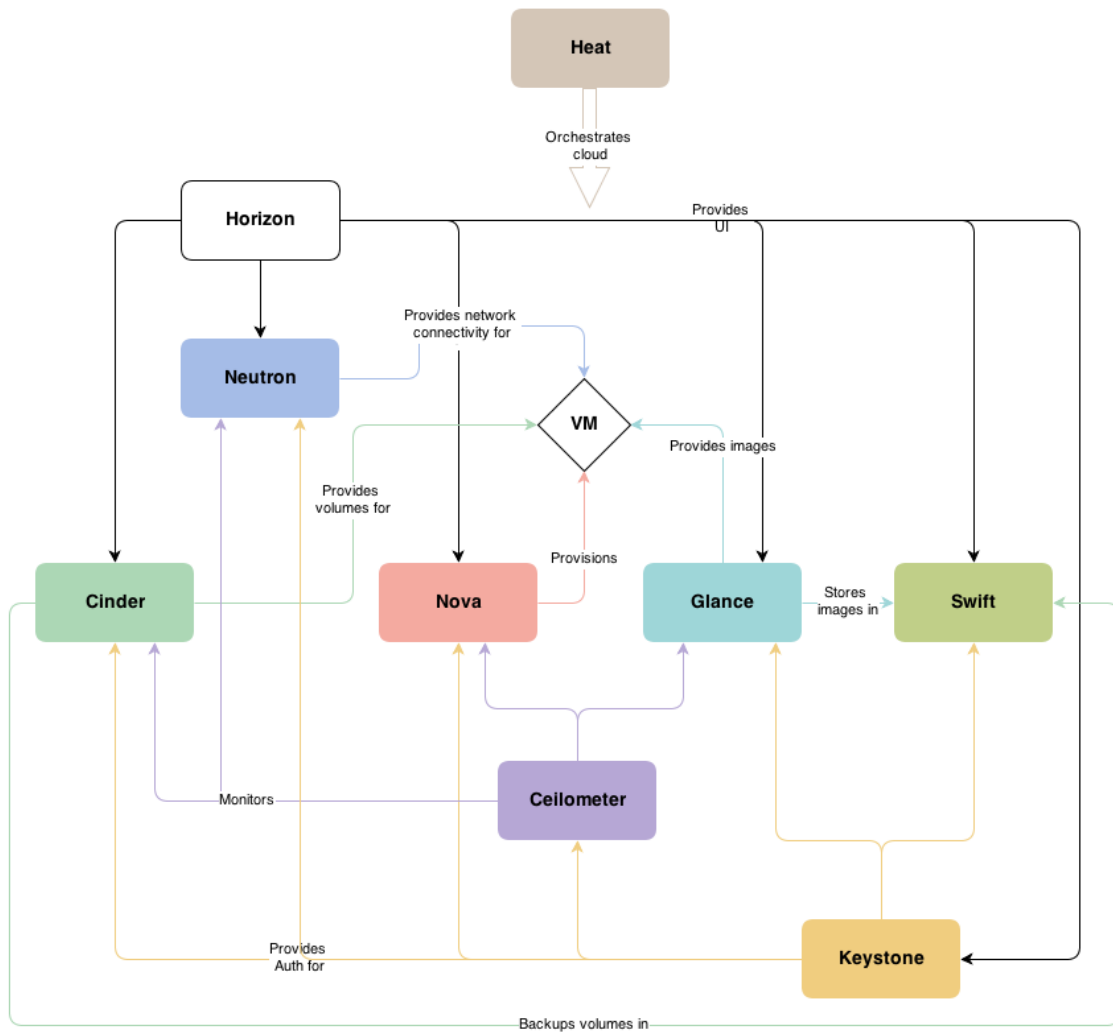


Figura 26: Esquema de serveis de l'OpenStack

## 8. Anàlisi i disseny del sistema

Les tasques d'anàlisi i disseny del sistema també es poden dividir en 3 etapes, degut als 3 escenaris que s'han muntat. Després d'haver establert els requisits del sistema a l'apartat 6 i haver definit les eines a utilitzar a l'apartat 7, cal realitzar el disseny de cadascun dels escenaris.

### 8.1 Escenari ESXi

Per poder fer funcionar l'OpenStack dins el servidor ESXi, el sistema (inicial) que calia muntar era el següent:

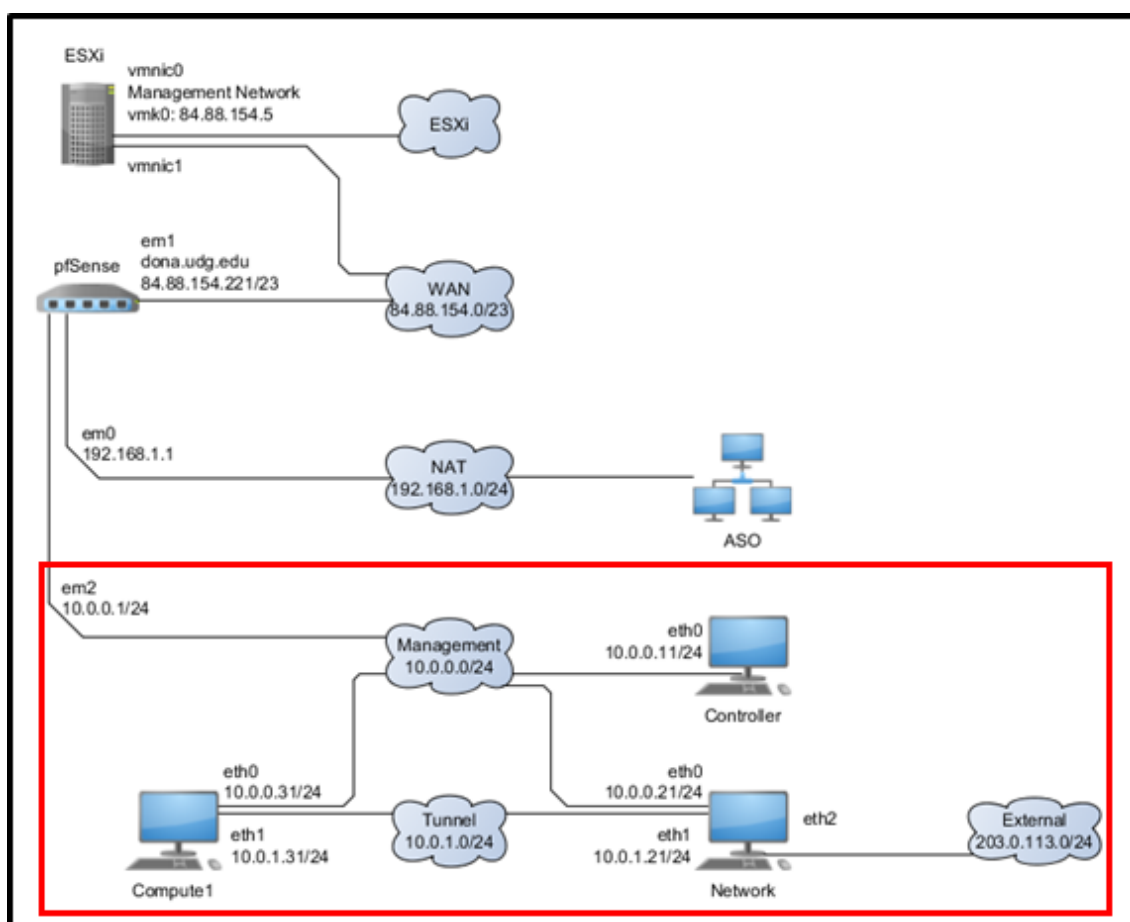


Figura 27: Topologia de l'escenari ESXi

Com es pot observar a la imatge, el requadre vermell es correspon a la infraestructura necessària per executar l'OpenStack, i la resta és el que ja hi havia prèviament al servidor, muntatge que fem servir al primer quadrimestre per l'assignatura d'Administració de Sistemes.

Les xarxes que ens interessin són la *Management*, *Tunnel* i *External*, que són sobre les que treballen els nodes de l'OpenStack. Així, les interfícies *eth0* de tots els nodes estan connectades a la xarxa *Management*, només el node *Compute1* i *Network* es connecten a la xarxa *Tunnel* a través de les seves interfícies *eth1* i, finalment, el node *Network* és qui es connecta a la xarxa *External* amb la seva interfície *eth2*. Cadascuna

d'aquestes xarxes la forma un *switch* virtual del servidor, que a la fase d'implementació s'explicarà com ha estat configurat.

D'altra banda, el pfSense també es connecta a la xarxa *Management*. D'aquesta manera, i amb les regles que es defineixin al pfSense, es podrà accedir de manera fàcil als nodes de l'OpenStack des de màquines que pertanyin a altres xarxes, així com connectar aquests a l'exterior (Internet), per les primeres configuracions, que necessitaran d'aquesta connexió.

## 8.2 Escenari PC personal

L'escenari dissenyat va ser el que segueix a continuació:

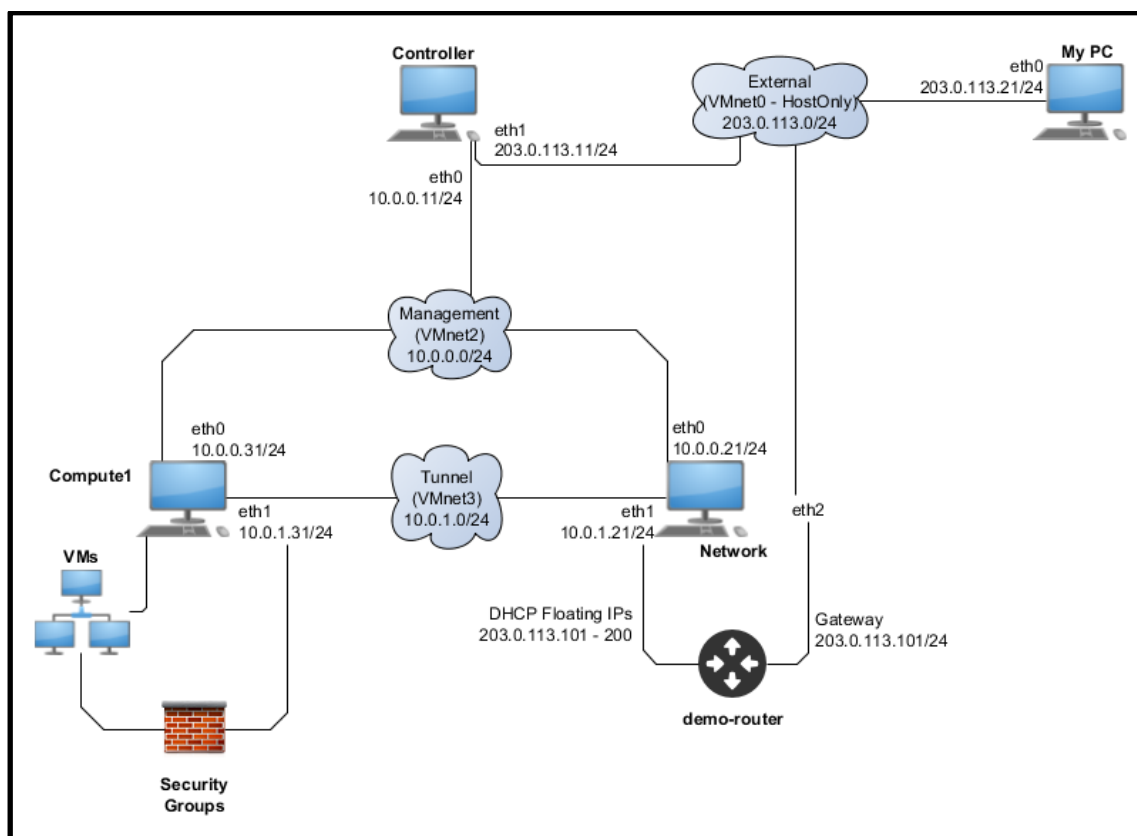


Figura 28: Topologia de l'escenari al PC personal

En aquest cas, tot el que formava part del muntatge de l'ESXi ja no existeix, i el que es manté són les 3 xarxes de l'OpenStack (*Management*, *Tunnel* i *External*) i ara el punt d'entrada al cloud es fa a través de la xarxa *External*, on s'hi connecten tan el *Controller* com el meu PC personal. A més també s'han especificat les màquines virtuals contingudes al node de computació, amb els seus *Security Groups* (regles de *firewall*) i la topologia de xarxa virtual que gestiona el node *Network* per connectar les màquines virtuals de l'OpenStack cap a l'exterior i de l'exterior cap a les màquines virtuals.

En aquest cas, totes les xarxes són virtuals també, fetes gràcies a l'VMware Player i la seva eina de configuració de xarxes virtuals.

### 8.3 Escenari final

L'escenari que es correspon amb les màquines virtuals és el següent:

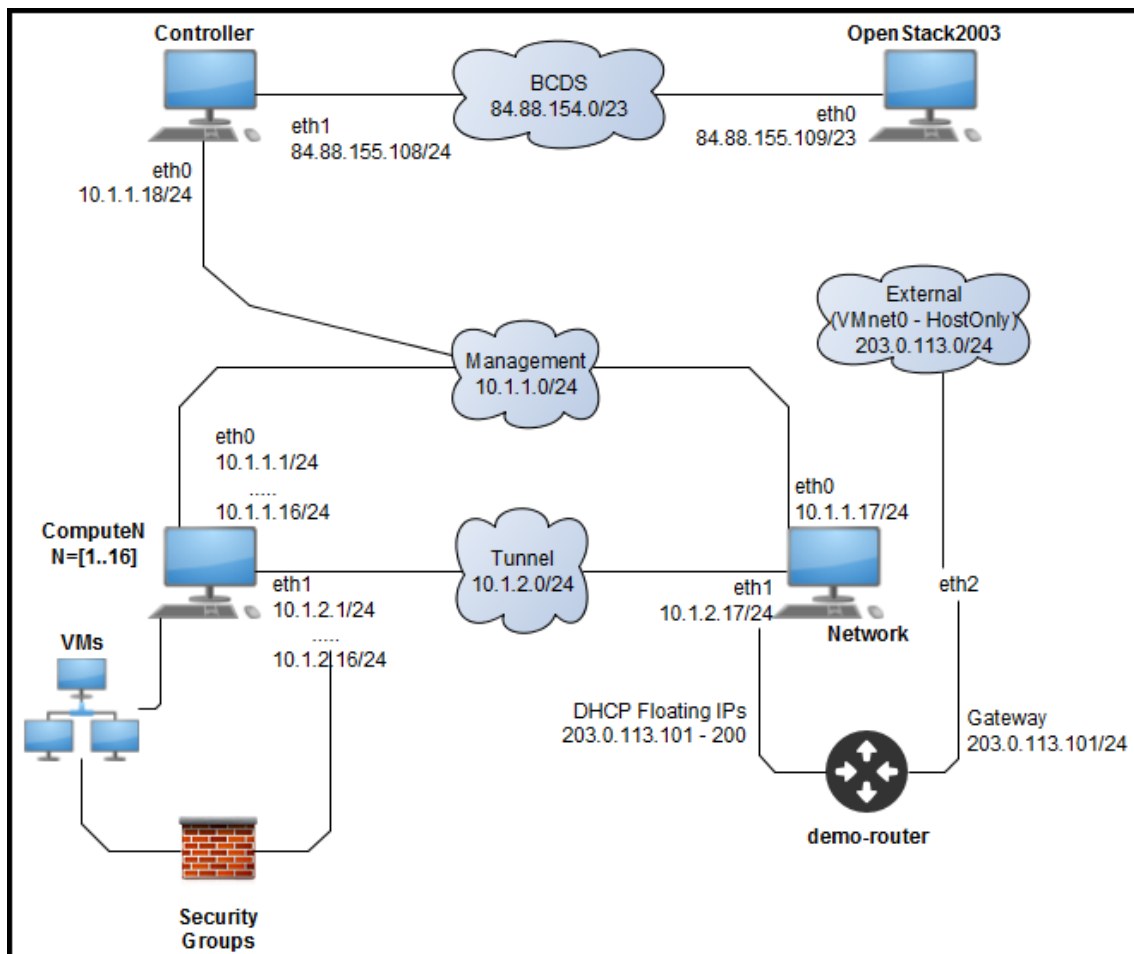


Figura 29: Topologia de l'escenari de l'aula III-04i

Com es pot observar, l'adreça de les xarxes ha canviat, donat que la configuració de la xarxa de l'aula és diferent. La connexió d'entrada disponible al *cloud* es fa a través del node *Controller* des de la seva interfície que connecta amb la xarxa de BCDS, per així tenir un accés directe a l'aplicació web que permet als seus usuaris administrar les seves màquines virtuals del *cloud*. A més a més, també es pot veure que apareix el servidor OpenStack2003 que controla l'arrencada i parada dels nodes OpenStack. Una altra observació a tenir en compte és que els nodes de computació es veuen representats pel node *ComputeN*, dels quals, com es pot comprovar, n'hi ha 16.

L'aula presenta una topologia de xarxa que està descrita a la següent taula:

Estació	Host	Guest	Private Net1	Private Net2
Comunicacions01	84.88.155.71/23	84.88.155.91/23	10.1.1.1/24	10.1.2.1/24
Comunicacions02	84.88.155.72/23	84.88.155.92/23	10.1.1.2/24	10.1.2.2/24
Comunicacions03	84.88.155.73/23	84.88.155.93/23	10.1.1.3/24	10.1.2.3/24
Comunicacions04	84.88.155.74/23	84.88.155.94/23	10.1.1.4/24	10.1.2.4/24
Comunicacions05	84.88.155.75/23	84.88.155.95/23	10.1.1.5/24	10.1.2.5/24
Comunicacions06	84.88.155.76/23	84.88.155.96/23	10.1.1.6/24	10.1.2.6/24
Comunicacions07	84.88.155.77/23	84.88.155.97/23	10.1.1.7/24	10.1.2.7/24
Comunicacions08	84.88.155.78/23	84.88.155.98/23	10.1.1.8/24	10.1.2.8/24
Comunicacions09	84.88.155.79/23	84.88.155.99/23	10.1.1.9/24	10.1.2.9/24
Comunicacions10	84.88.155.80/23	84.88.155.100/23	10.1.1.10/24	10.1.2.10/24
Comunicacions11	84.88.155.81/23	84.88.155.101/23	10.1.1.11/24	10.1.2.11/24
Comunicacions12	84.88.155.82/23	84.88.155.102/23	10.1.1.12/24	10.1.2.12/24
Comunicacions13	84.88.155.83/23	84.88.155.103/23	10.1.1.13/24	10.1.2.13/24
Comunicacions14	84.88.155.84/23	84.88.155.104/23	10.1.1.14/24	10.1.2.14/24
Comunicacions15	84.88.155.85/23	84.88.155.105/23	10.1.1.15/24	10.1.2.15/24
Comunicacions16	84.88.155.86/23	84.88.155.106/23	10.1.1.16/24	10.1.2.16/24
Comunicacions17	84.88.155.87/23	84.88.155.107/23	10.1.1.17/24	10.1.2.17/24
Comunicacions18	84.88.155.88/23	84.88.155.108/23	10.1.1.18/24	10.1.2.18/24
Comunicacions19	84.88.155.89/23	84.88.155.109/23	10.1.1.19/24	10.1.2.19/24

Taula 1: Mapa d'IPs de l'aula.

La primera columna (Estació) de la taula anterior representa els noms de *host* que tenen les màquines de l'aula. Abans de realitzar el muntatge del *cloud*, ja estaven assignats des de Comunicacions01 fins a Comunicacions18. Després d'haver afegit una nova màquina que les administri, aquesta s'anomenarà Comunicacions19 que és el nom de *host* que quedava lliure.

La segona columna (*Host*) representa les IPs que s'han d'assignar a les màquines reals. Novament, estan assignades les 18 primeres, quedant lliure la darrera (84.88.155.89/23) per Comunicacions19, l'estació que contindrà el servidor.

La tercera columna (*Guest*) són les IPs públiques que poden ser assignades a cadascuna de les màquines virtuals que existeixin en aquesta topologia. Cal tenir en compte que cada màquina real (ComunicacionsXX) només disposa d'una IP pública per màquines virtuals. Això vol dir que si hi ha varies màquines virtuals dins una mateixa estació, només una d'elles pot fer servir aquesta adreça IP.

En el cas del disseny realitzat, aquesta limitació no ha suposat cap problema, donat que les úniques màquines virtuals que necessitaven una IP pública eren el *Controller* de l'Openstack i el servidor que les administra, amb nom de *host OpenStack2003*; contingudes cadascuna d'elles en màquines reals diferents, Comunicacions18 i Comunicacions19, respectivament.

La quarta columna (*Private Net1*) es correspon a les IPs d'una xarxa privada que no tenia ús fins ara. Gràcies a l'existència d'aquesta xarxa, vaig poder adjudicar-li la funció de xarxa *Management* de l'OpenStack.

La darrera columna (*Private Net2*) es va haver de sol·licitar als administradors de xarxa, ja que no existia. És la xarxa que s'ha fet servir com a xarxa *Tunnel* de l'OpenStack.

Totes aquestes xarxes s'han fet funcionar sota una única xarxa física, simplement el *switch* que la gestiona està configurat per poder treballar amb les 4 xarxes. En el següent gràfic es pot veure a nivell de hosts reals i de connexions amb el *switch*. Cada *host* té a la seva esquerra la IP real que té, i a la dreta les IPs de les màquines virtuals que conté, quedant reflectit així, el trànsit que passa per cada interfície de cada *host*.

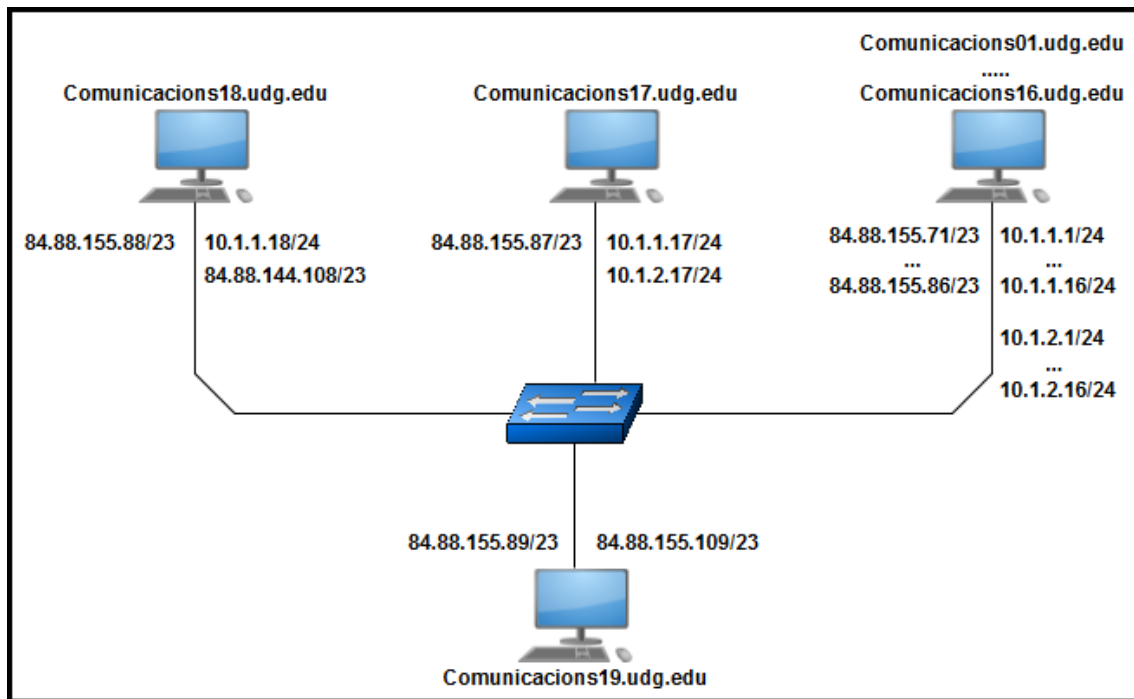


Figura 30: IPs que utilitza cada màquina

Durant l'apartat d'implementació s'explicarà amb detall com ha de ser la configuració tan de les estacions com de les màquines virtuals per poder aconseguir aquest escenari.

En el següent esquema es pot veure com s'han repartit les màquines virtuals a través de les estacions que conformen l'aula. La repartició dels nodes de computació i el de control s'ha fet d'aquesta manera, 1 màquina virtual per 1 màquina real, perquè les màquines virtuals aprofiten la totalitat de recursos de la màquina real. Pel que fa al node *Network*, no consumeix tots els recursos de la màquina real, però com que aquesta només disposa d'una sola IP per rang, només es pot posar una màquina virtual. Pel que fa a la màquina que conté el servidor d'administració (Comunicacions 19 que conté OpenStack2003), ha estat així perquè s'aprofiten els recursos de la màquina i pel tema d'assignacions d'una sola IP per rang i estació.



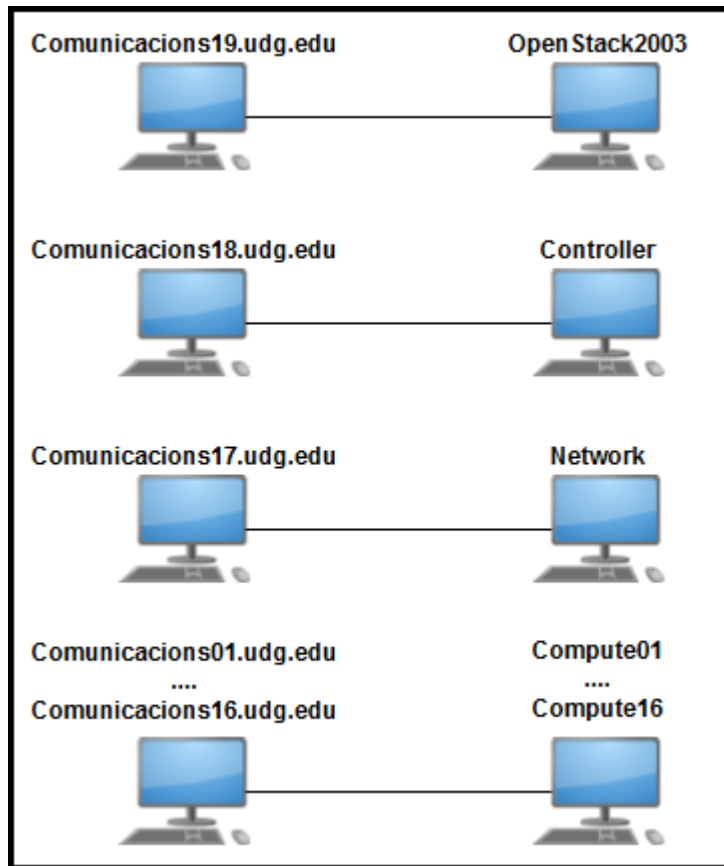


Figura 31: VMs que conté cada estació

## 9. Implementació i proves

Al llarg de la memòria s'ha anat parlant que s'han muntat 3 escenaris, per tant, aquest apartat també està enfocat com l'anterior, 3 apartats que representen els escenaris. D'aquesta manera quedaran més clares cadascuna de les implementacions i proves realitzades

### 9.1 Escenari ESXi

Per començar a muntar aquest escenari, cal connectar-se a l'*vSphere* del servidor ESXi, i això es fa amb el client:

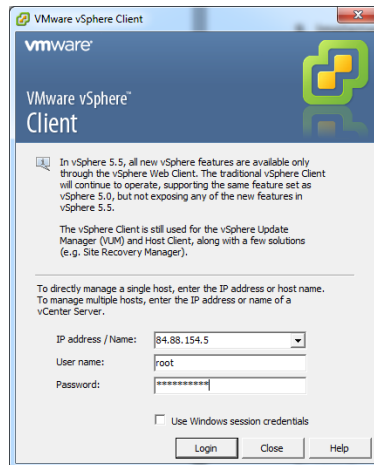


Figura 32: Client vSphere

Un cop connectat a la consola d'administració, el primer pas que vaig fer va ser crear 3 màquines virtuals que representessin cadascun dels nodes (*Controller*, *Network*, *Compute1*) amb les característiques recomanades (especificades a l'apartat 6, de requisits del sistema):

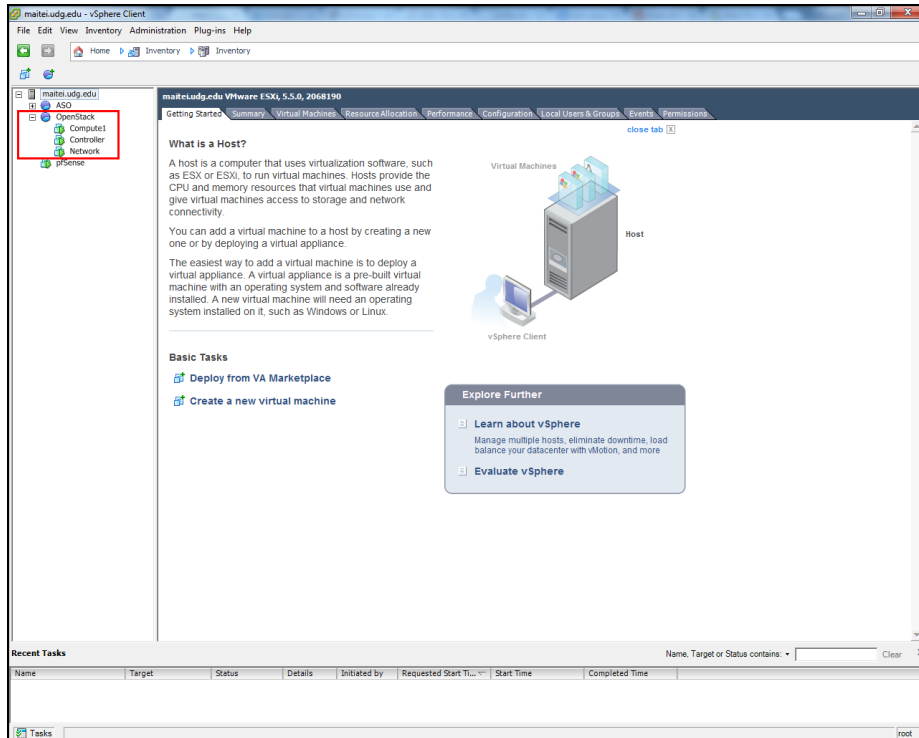


Figura 33: Consola d'administració de l'vSphere

Seguidament calia afegir 3 *switchs* virtuals a l'ESXi per les xarxes *Management*, *Tunnel* i *External*. Per fer-ho només cal prémer el vincle "Add Networking..." i seguir l'assistent que apareix.

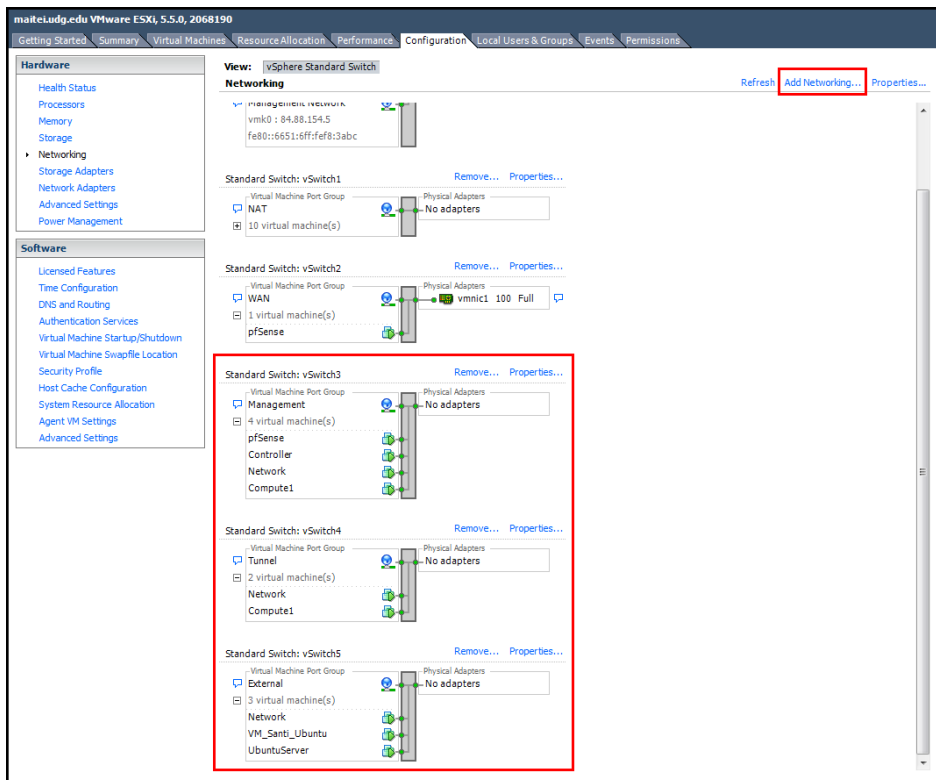


Figura 34: Topologia de les xarxes virtuals a l'ESXi

El *switch* que representa la xarxa *External* necessita una configuració addicional, que és habilitar el mode *promiscu*. Com més endavant es veurà, el node que s'hi connectarà serà el *Network* a través de la seva interfície *eth2*, la qual no tindrà configuració IP, degut a que ha d'atendre tot el trànsit i no només el que vagi dirigit per ella, ja que aquesta interfície serà la que connecti les màquines virtuals de dins del cloud cap a l'exterior motiu pel qual:

- S'han de capturar tots els paquets per analitzar-los, ja que no seran dirigits cap a *eth2*, sinó cap a les interfícies virtuals que depenguin d'aquesta.
- No ha de tenir configuració IP, ja que no hi haurà cap paquet dirigit cap a aquesta interfície, representant el rol de "interfície d'interfícies".

El mode de treball *promiscu* s'ha de configurar a dos llocs diferents: a l'ESXi i al fitxer de configuració `/etc/network/interfaces` del node *Network*. Per l'ESXi, en primer lloc cal anar a les propietats del *switch External*:

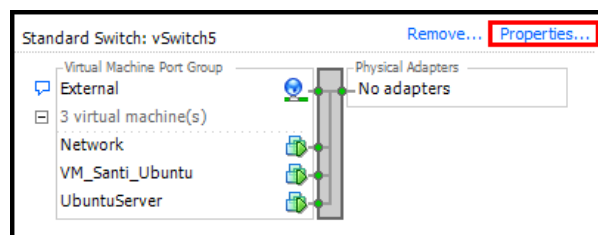


Figura 35: Switch virtual External

Seguidament el botó *Edit...*

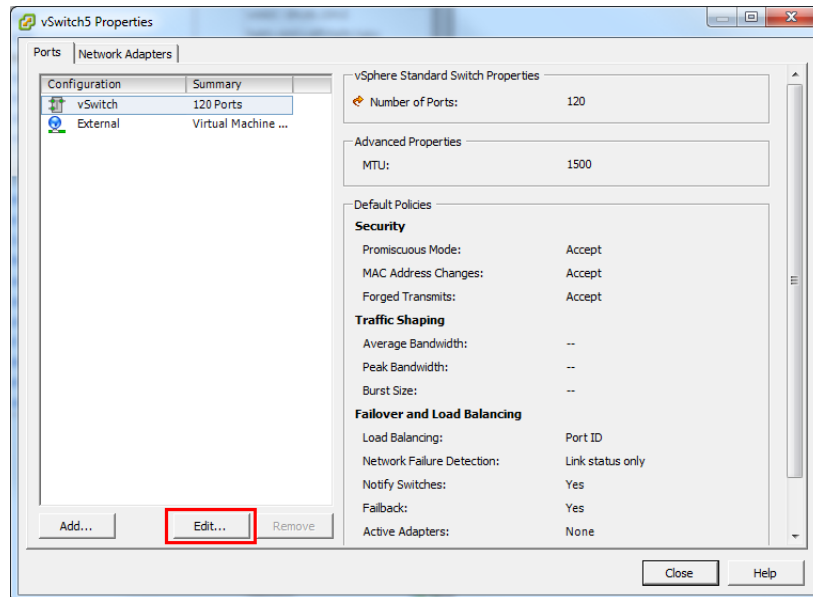


Figura 36: Propietats del switch External

Pestanya *Security* i desplegar el menú *Promiscuous Mode* per triar l'*Accept*.

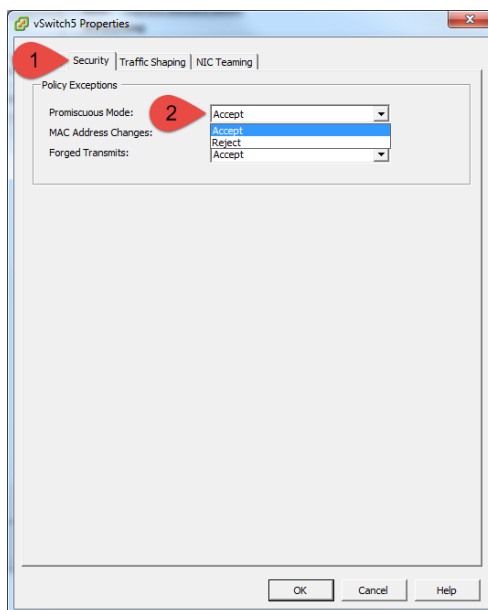


Figura 37: Propietats detallades del switch External

Premem OK i aquest *switch* ja admet el mode promiscu per les interfícies de les màquines que s'hi connectin.

El següent pas és instal·lar l'Ubuntu 14.04 LTS a cadascuna de les màquines virtuals i realitzar la configuració de xarxa adient per cadascuna d'elles.

Per la màquina "Controller" se li ha posat aquest nom de host i a més se li ha dit on trobar la VM "Network" i la "Compute1":

```
santi@Controller:~$ cat /etc/hostname
Controller
santi@Controller:~$
santi@Controller:~$
santi@Controller:~$ cat /etc/hosts
127.0.0.1    localhost
#127.0.1.1  Controller

# OpenStack nodes
10.0.0.11   Controller
10.0.0.21   Network
10.0.0.31   Compute1

# The following lines are desirable fo
::1        localhost ip6-localhost ip6-lc
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

I la seva configuració de xarxa és:

```

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# Management network
auto eth0
iface eth0 inet static
address 10.0.0.11
netmask 255.255.255.0
gateway 10.0.0.1
dns-nameservers 10.0.0.1

```

Per la màquina "Network" se li ha posat aquest nom de host i a més se li ha dit on trobar la VM "Controller" i la "Compute1":

```

santi@Network:~$ cat /etc/hostname
Network
santi@Network:~$
santi@Network:~$
santi@Network:~$ cat /etc/hosts
127.0.0.1    localhost
#127.0.1.1  Network

# OpenStack nodes
10.0.0.11   Controller
10.0.0.21   Network
10.0.0.31   Compute1

# The following lines are desirable
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

```

I la seva configuració de xarxa és:

```

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# Management network interface
auto eth0
iface eth0 inet static
address 10.0.0.21
netmask 255.255.255.0
gateway 10.0.0.1
dns-nameservers 10.0.0.1

# Tunnel network interface
auto eth1
iface eth1 inet static
address 10.0.1.21
netmask 255.255.255.0

# External network interface
# Special configuration without IP address assigned
auto eth2
iface eth2 inet manual
up ip link set $IFACE promisc on
up ip link set dev $IFACE up
down ip link set $IFACE promisc off
down ip link set dev $IFACE down

```

Per la màquina "Compute1" se li ha posat aquest nom de host i a més se li ha dit on trobar la VM "Controller" i la "Network":

```
santi@Compute1:~$ cat /etc/hostname
Compute1
santi@Compute1:~$
santi@Compute1:~$
santi@Compute1:~$ cat /etc/hosts
127.0.0.1    localhost
#127.0.1.1   Compute1

# OpenStack nodes
10.0.0.11    Controller
10.0.0.21    Network
10.0.0.31    Compute1

# The following lines are desirable
::1    localhost ip6-localhost ip6
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

I la seva configuració de xarxa és:

```
# This file describes the network
# and how to activate them. For n

# The loopback network interface
auto lo
iface lo inet loopback

# Management network interface
auto eth0
iface eth0 inet static
address 10.0.0.31
netmask 255.255.255.0
gateway 10.0.0.1
dns-nameservers 10.0.0.1

# Tunnel network interface
auto eth1
iface eth1 inet static
address 10.0.1.31
netmask 255.255.255.0
```

Tal i com es pot observar a la configuració de xarxa de les tres màquines virtuals, a la interfície corresponent a la xarxa "Management" tenen associada una *gateway*, la 10.0.0.1. El fet és que les màquines virtuals necessiten connexió a Internet (per poder baixar els paquets de l'OpenStack, un cop instal·lats ja no caldrà), i ha de ser a través d'aquesta xarxa. Per aconseguir-ho, el que he hagut de fer ha estat afegir una interfície de xarxa nova a la VM pfSense en primer lloc, editant les preferències:

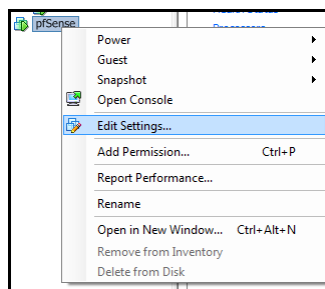


Figura 38: Acció de botó dret

I afegint-la:

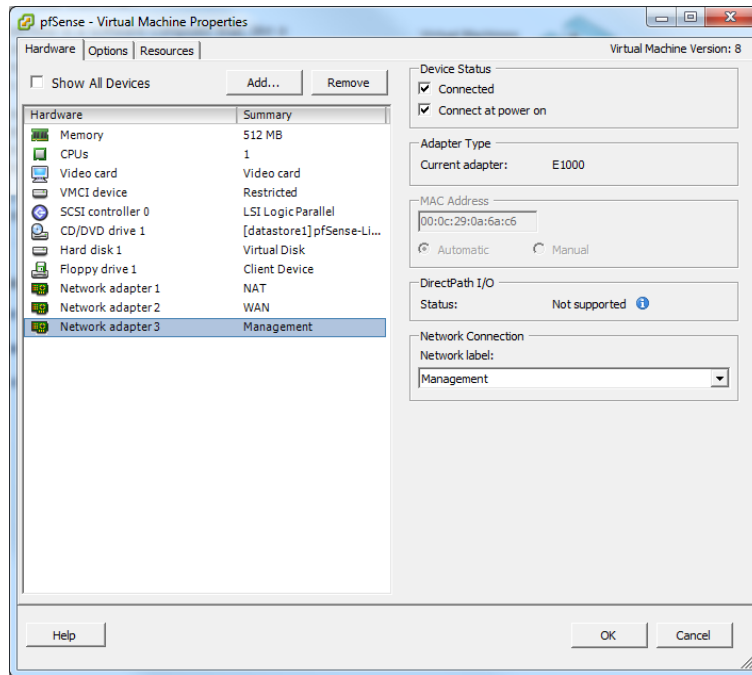


Figura 39: Propietats de la VM

Un cop afegit el hardware necessari a la VM, calia configurar-la al pfSense. Per començar calia afegir-la i assignar-li un nom. Per fer-ho, cal anar al menú *Interfaces*→(*assign*). Un cop allà, es pot veure que s'ha activat un botó a la dreta amb el símbol '+', es pressiona i llavors només cal assignar-li la nova interfície i el nom que es vulgui posar, en aquest cas ha quedat així:

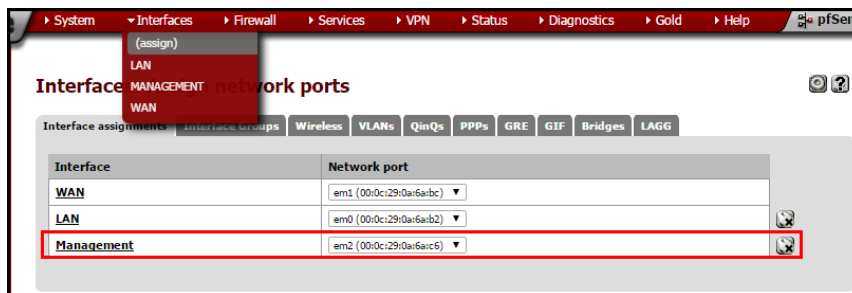


Figura 40: Configuració d'interfícies del pfSense

Cal tenir en compte que en el menú que es desplega des de "*Interfaces*", abans d'afegir la interfície, aquesta no hi apareix. Un cop afegida, per poder configurar-la, només cal tornar al menú "*Interfaces*" i seleccionar-la. La configuració realitzada és la corresponent als requadres vermells de la següent figura:



## Interfaces: Management

**General configuration**

Enable  **Enable Interface**

Description   
Enter a description (name) for the interface here.

IPv4 Configuration Type

IPv6 Configuration Type

MAC address   
Insert my local MAC address  
This field can be used to modify ("spoof") the MAC address of this interface (may be required with some cable connections)  
Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank

MTU   
If you leave this field blank, the adapter's default MTU will be used. This is typically 1500 bytes but can vary in some circumstances.

MSS   
If you enter a value in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect.

Speed and duplex  - Show advanced option

**Static IPv4 configuration**

IPv4 address  /

IPv4 Upstream Gateway  - or add a new one.  
If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the link above.  
On local LANs the upstream gateway should be "none".

**Track IPv6 Interface**

IPv6 Interface

This selects the dynamic IPv6 WAN interface to track for configuration

IPv6 Prefix ID   
The value in this field is the (Delegated) IPv6 prefix id. This determines the configurable network ID based on the dynamic IPv6 connection  
Enter a **hexadecimal** value between 0 and 0 here, default value is 0.

**Private networks**

**Block private networks**  
When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). You should generally leave this option turned on, unless your WAN network lies in such a private address space, too.

**Block bogon networks**  
When set, this option blocks traffic from IP addresses that are reserved (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and obviously should not appear as the source address in any packets you receive.

Note: The update frequency can be changed under System->Advanced Firewall/NAT settings.

Figura 41: Configuració de la interfície Management

El següent pas és anar al següent menú:

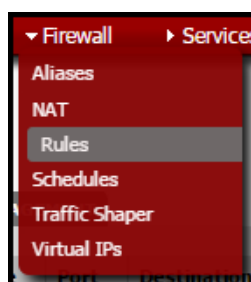


Figura 42: Regles del firewall

I a la pestanya "MANAGEMENT" afegir-hi les següents regles:

ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
	IPv4 *	MANAGEMENT net	*	*	*	*	none		Default allow MANAGEMENT to any rule
	IPv6 *	MANAGEMENT net	*	*	*	*	none		Default allow LAN IPv6 to any rule

pass  
 pass (disabled)
  block  
 block (disabled)
  reject  
 reject (disabled)
  log  
 log (disabled)

**Hint:**  
 Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

Figura 43: Regles del firewall per la xarxa Management

Després de tota la configuració de xarxa, per anar sobre segur, caldria reiniciar les màquines. Quan ja estan totes a punt, cal fer les proves de connectivitat necessàries per comprovar que el muntatge s'hagi realitzat correctament; bàsicament cal fer *ping* allà on toqui. Si tots els *pings* han funcionat, l'esquema lògic que ha quedat és el següent:

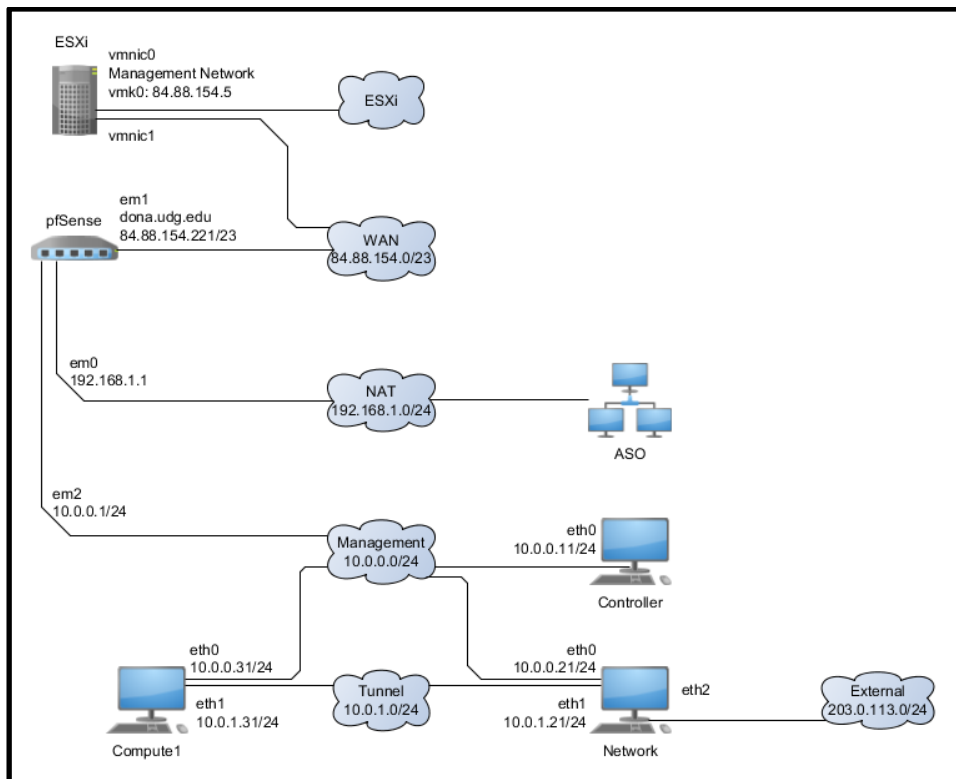


Figura 44: Escenari ESXi

## 9.1.2 Preparació del Software bàsic de les VMs

### Network Time Protocol (NTP)

El següent pas es tracta de configurar el servei NTP a les tres VMs per poder sincronitzar els diferents serveis implicats en la gestió i el funcionament del Cloud. Per fer-ho cal en primer lloc definir la màquina que es sincronitzarà amb un servei NTP extern i la que servirà per sincronitzar la resta. En aquest cas, per analogia, el més normal és triar el **Controller**, sent el **pfSense** la màquina que li facilitarà la sincronització a ella, ja que disposava d'un servei NTP funcionant. **Network** i **Compute1** es sincronitzaran amb el **Controller**.

Per començar, cal instal·lar el servei a cadascuna de les màquines:

```
santi@Controller:~$ sudo apt-get install ntp
```

I després editar el fitxer de configuració de cadascuna de les màquines:

```
santi@Controller:~$ sudo vim /etc/ntp.conf
```

De la següent manera; **Controller**:

```
# Specify one or more NTP servers.
server pfsense.dona.udg.edu iburst
server 0.pfsense.pool.ntp.org iburst

# Use servers from the NTP Pool Project
# on 2011-02-08 (LP: #104525). See http://www.ubuntu.com/ntp
# more information.
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
```

**Network** i **Compute1**:

```
# Specify one or more NTP servers.
server controller iburst
```

Un cop fet això, només cal reiniciar el servei NTP a totes les màquines:

```
santi@Network:~$ sudo service ntp restart
```

Finalment cal comprovar que l'NTP funciona. En primer lloc cal fer al **Controller**:

```
santi@Controller:~$ ntpq -c peers
=====
      remote           refid      st t when poll reach  delay  offset jitter
=====
-pfSense.dona.ud 81.19.96.148  3 u  815 1024 377   0.360  -18.418  0.596
+static-Z1.herco 14.234.237.209 2 u  189 1024 377  34.487  -10.969  2.967
-ns1.indalweb.n 193.190.230.65 2 u  803 1024 377  45.175  -7.656  6.818
*84.88.69.32     193.67.79.202 2 u  192 1024 377   4.732  -9.922  2.538
+vps01.roethof.n 192.87.36.4   2 u 1173 1024 366  51.205 -10.829  5.615
- juniperberry.ca 192.93.2.20   2 u  947 1024 377  38.707  -9.517  0.794
```

I observar que s'hagi afegit el servidor NTP que havia configurat. El següent pas és també al **Controller**:

```
santi@Controller:~$ ntpq -c assoc
```

ind	assid	status	conf	reach	auth	condition	last_event	cnt
1	38176	933a	yes	yes	none	outlyer	sys_peer	3
2	38177	941a	yes	yes	none	candidate	sys_peer	1
3	38178	931a	yes	yes	none	outlyer	sys_peer	1
4	38179	961a	yes	yes	none	sys.peer	sys_peer	1
5	38180	943a	yes	yes	none	candidate	sys_peer	3
6	38181	931a	yes	yes	none	outlyer	sys_peer	1

Observi's que s'ha produït la "condition" "sys.peer" per almenys un servidor. Ara ja només cal comprovar els altres dos nodes (**Network** i **Compute1**):

```
santi@Compute1:~$ ntpq -c peers
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*Controller	84.88.69.32	3	u	13	64	1	0.267	9.377	0.139

```
santi@Compute1:~$ ntpq -c assoc
```

ind	assid	status	conf	reach	auth	condition	last_event	cnt
1	10234	963a	yes	yes	none	sys.peer	sys_peer	3

## Definició del repositori i instal·lació de paquets

Les següents accions s'han de fer als tres nodes. Primer cal instal·lar el paquet "ubuntu-cloud-keyring":

```
santi@Network:~$ sudo apt-get install ubuntu-cloud-keyring
```

Seguidament cal crear un fitxer on s'hi definirà el repositori nou:

```
santi@Network:~$ sudo touch /etc/apt/sources.list.d/cloudarchive-juno.list
```

El contingut del fitxer ha de ser:

```
deb http://ubuntu-cloud.archive.canonical.com/ubuntu trusty-updates/juno main
```

Finalment, ja es poden instal·lar els paquets i actualitzar el sistema (cal reiniciar):

```
santi@Network:~$ sudo apt-get update && sudo apt-get dist-upgrade
```

## Base de dades

La majoria de serveis de l'Openstack necessiten bases de dades SQL per guardar-hi informació, allotjades al node **Controller**. Passos a seguir, des del node Controller, és clar:

1. Instal·lar el servidor de la base de dades (*MariaDB-Server*) i la llibreria *Python-MySQLDB*:

```
santi@Controller:~$ sudo apt-get install mariadb-server python-mysqldb
```

2. Assignar un password per l'administrador de la BD:

```
Configuring mariadb-server-5.5
While not mandatory, it is highly recommended that you set a password for the MariaDB
administrative "root" user.

If this field is left blank, the password will not be changed.

New password for the MariaDB "root" user:
*****
<Ok>
```

3. Editar el fitxer de configuració del MySQL:

```
santi@Controller:~$ sudo vim /etc/mysql/my.cnf
```

- A la secció **[mysqld]** cal posar la IP corresponent a la interfície de la xarxa **Management** (10.0.0.11) al paràmetre *bind-address*:

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address          = 127.0.0.1
#
# To receive queries via Management network, the bind-address
# must be the corresponding IP (eth0 = 10.0.0.11)
bind-address          = 10.0.0.11
```

- A la mateixa secció, cal configurar el paràmetres de la següent manera (opcions útils a més d'opcions per a la codificació en UTF-8):

```
#
# Useful options and UTF-8 character set
#
default-storage-engine = innodb
innodb_file_per_table
collation-server        = utf8_general_ci
init-connect            = 'SET NAMES utf8'
character-set-server    = utf8
```

4. Reiniciar el servei:

```
santi@Controller:~$ sudo service mysql restart
```

5. Fer-lo segur:

```
santi@Controller:~$ sudo mysql_secure_installation
```

Aquest darrer pas es tracta d'un assistent que anirà realitzant preguntes de com ha de quedar configurada la base de dades:

- Primer demana la contrasenya de *root* que hem posat abans:

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

- Seguidament demana si es vol canviar la contrasenya; trio no canviar-la, no cal:

```
Change the root password? [Y/n] n
... skipping.
```

- Després diu si vull que s'eliminïn els usuaris anònims; trio que els elimini ja que és perillós que qualsevol pugui entrar a la base de dades:

```
Remove anonymous users? [Y/n] y
... Success!
```

- No s'ha de permetre que es pugui fer *login* amb l'usuari *root* remotament:

```
Disallow root login remotely? [Y/n] y
... Success!
```

- Cal eliminar la base de dades "test", ja que tothom hi pot accedir i només és per propòsits de *testing*:

```
Remove test database and access to it? [Y/n] y
- Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist
... Failed! Not critical, keep moving...
- Removing privileges on test database...
... Success!
```

- Cal garantir que els canvis que s'han produït es facin efectius recarregant els privilegis a les taules:

```
Reload privilege tables now? [Y/n] y
... Success!
```

- Finalment, s'ha acabat d'assegurar el servei:

```
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

### Servei de missatgeria (Message broker, intercanviador de missatges)

L'OpenStack necessita un servei que s'encarregui de l'intercanvi de missatges entre els diferents serveis, per coordinar les operacions i la informació del seu estat. Aquest servei també el farà córrer el node **Controller**. El que farà servir és el *RabbitMQ*.

En primer lloc cal instal·lar el servei:

```
santi@Controller:~$ sudo apt-get install rabbitmq-server
```

Per utilitzar el servei cal disposar de credencials. Per defecte, el *RabbitMQ* disposa d'un usuari per defecte anomenat *guest* amb la contrasenya també *guest*. Per fer el servei més segur hi ha diverses opcions:

- Canviar la contrasenya de l'usuari *guest* i fer-lo servir com a credencials úniques per a tots els serveis que hagin d'utilitzar el servei d'intercanvi de missatges.
- Crear un usuari nou amb una contrasenya nova i que l'utilitzin els serveis que el necessitin.
- Crear un usuari per cada servei.

Com que de moment estic de proves, trio la primera opció que és la que també segueix el manual. La contrasenya que poso és la mateixa que per l'administració de la base de dades:

```
santi@Controller:~$ sudo rabbitmqctl change_password guest 20openstack15
[sudo] password for santi:
Changing password for user "guest" ...
...done.
```

El mapa lògic de serveis és el següent:

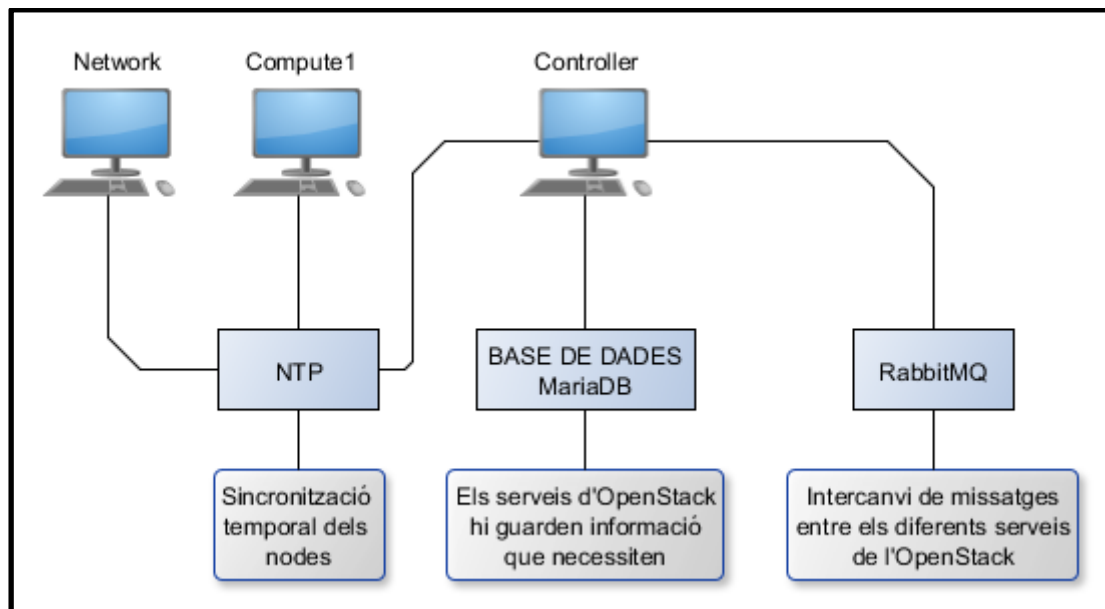


Figura 45: Conjunt de serveis bàsics

## 9.1.2 Instal·lació i configuració dels serveis OpenStack

### 9.1.2.1 Identity service (Keystone)

#### Instal·lació i configuració

##### *Prerequisites*

Es necessita una base de dades i un *token* d'administració com a prerequisit per poder realitzar la instal·lació i configuració del servei.

En primer lloc cal crear la base de dades:

1. Des del node *Controller*, fer *login* al servidor de base de dades (*MariaDB*) amb el client d'accés a la base de dades (*python mysql*):

```
santi@Controller:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 28
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

2. Crear una base de dades anomenada **keystone**:

```
MariaDB [(none)]> CREATE DATABASE keystone;
Query OK, 1 row affected (0.00 sec)
```

3. Donar accés a la nova base de dades a l'usuari *keystone*:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY '20keystone15';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY '20keystone15';
Query OK, 0 rows affected (0.00 sec)
```

En segon lloc cal generar un valor aleatori per fer servir com a **token** d'administració (*token\_admin*) durant la configuració inicial:

```
santi@Controller:~$ openssl rand -hex 10
46b2e7eeacc97e3aac9d
```

#### *Instal·lació i configuració*

Passos a seguir:

1. Instal·lació dels paquets necessaris:

```
santi@Controller:~$ sudo apt-get install keystone python-keystoneclient
```

2. Editar el fitxer de configuració del **keystone** (*/etc/keystone/keystone.conf*):

```
santi@Controller:~$ sudo vim /etc/keystone/keystone.conf
```



- a) Afegir el **token\_admin** prèviament generat a la secció [DEFAULT]:

```
[DEFAULT]
#
# Options defined in keystone
#
# A "shared secret" that can be used to bootstrap Keystone.
# This "token" does not represent a user, and carries no
# explicit authorization. To disable in production (highly
# recommended), remove AdminTokenAuthMiddleware from your
# paste application pipelines (for example, in keystone-
# paste.ini). (string value)
admin_token=46b2e7eeacc97e3aac9d
```

- b) A la secció [database] cal configurar l'accés a la base de dades:

```
[database]
#
# Options defined in oslo.db
#
# The file name to use with SQLite. (string value)
#sqlite_db=oslo.sqlite
# If True, SQLite uses synchronous mode. (boolean value)
#sqlite_synchronous=true
# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend=sqlalchemy
# The SQLAlchemy connection string to use to connect to the
# database. (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql://keystone:20keystone15@controller/keystone
```

- c) A la secció [token] s'ha de configurar l'UUID del proveïdor de **tokens** i el driver SQL:

```
[token]
#
# Options defined in keystone
#
# External auth mechanisms that should add bind information to
# token, e.g., kerberos,x509. (list value)
#bind=
# Enforcement policy on tokens presented to Keystone with bind
# information. One of disabled, permissive, strict, required
# or a specifically required bind mode, e.g., kerberos or x509
# to require binding to that authentication. (string value)
#enforce_token_bind=permissive
# Amount of time a token should remain valid (in seconds).
# (integer value)
#expiration=3600
# Controls the token construction, validation, and revocation
# operations. Core providers are
# "keystone.token.providers.[pkiz|pki|uuid].Provider". The
# default provider is uuid. (string value)
provider = keystone.token.providers.uuid.Provider
# Token persistence backend driver. (string value)
driver = keystone.token.persistence.backends.sql.Token
```

- d) A la secció [revoke] cal configurar el driver de revocació SQL:

```
[revoke]
#
# Options defined in keystone
#
# An implementation of the backend for persisting revocation
# events. (string value)
driver = keystone.contrib.revoke.backends.sql.Revoke
```

- e) (Opcional) Per tenir informació de possibles problemes, habilitar el *verbose* a la secció [DEFAULT]:

```
# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose = true
```

3. Emplenar la base de dades del servei **Identity (keystone)**:

```
santi@Controller:~$ su -s /bin/sh -c "keystone-manage db_sync" keystone
```

Aquesta primera comanda, que és la del manual, no funciona, ja que l'usuari *keystone* no té contrasenya (en cap moment li he posat) i quan s'executa, el sistema la demana. Aquest usuari sí que figura a */etc/passwd*, sense *shell* ni *home*:

```
keystone:x:107:115::/var/lib/keystone:/bin/false
```

Com que no sé si es poden provocar errors per canviar la contrasenya d'aquest usuari, el que faré és realitzar la comanda amb les meves credencials de *root* i ja està, cosa que sí que ha funcionat:

```
santi@Controller:~$ sudo keystone-manage db_sync
```

Després d'haver-me fixat en comandes posteriors, he pogut veure que aquestes van precedides pel caràcter sostingut (#), cosa que vol dir que s'executen totes amb credencials de *root*. Així, aquesta comanda també es pot executar com:

```
santi@Controller:~$ sudo su -s /bin/sh -c "keystone-manage db_sync" keystone
```

### Finalització de la instal·lació

1. Reiniciar el servei **Identity**:

```
santi@Controller:~$ sudo service keystone restart
keystone stop/waiting
keystone start/running, process 6232
```

2. A la instal·lació del *keystone* per l'Ubuntu es crea una base de dades *SQLite*, com que aquesta configuració fa servir un servidor de bases de dades SQL, es pot eliminar el fitxer de la base de dades:

```
santi@Controller:~$ sudo rm -f /var/lib/keystone/keystone.db
```

3. Per defecte, el servei **Identity** emmagatzema *tokens* expirats a la base de dades indefinidament. Com que això no cal, es poden anar eliminant periòdicament, afegint una entrada al **cron**:

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
@hourly root /usr/bin/keystone-manage token_flush >/var/log/keystone/keystone-tokenflush.log 2>&1
```

### Creació de contenidors (*tenants*), usuaris i rols

Ara toca crear el que el títol indica. S'ha d'utilitzar el *token* d'administració que s'ha generat anteriorment i configurar la localització del servei **Identity** abans de començar a executar comandes **keystone**. Per realitzar aquestes configuracions, es farà mitjançant variables d'entorn temporals, tot i que també es podria realitzar passant-ho per paràmetre (*--os-token* i *--os-endpoint*) cada vegada a la comanda **keystone**. Com

que no és pràctic, millor definir les variables d'entorn. Cal tenir en compte que aquestes, al ser temporals, es perden cada cop que es reinicia la màquina (per fer-les servir s'han de tornar a definir).

### Prerequisits (variables d'entorn)

- Configuració del *token* d'administració:

```
santi@Controller:~$ export OS_SERVICE_TOKEN=46b2e7eeacc97e3aac9d
```

- Configuració de l'*endpoint* (localització del servei):

```
santi@Controller:~$ export OS_SERVICE_ENDPOINT=http://controller:35357/v2.0
```

### Creació

1. Crear un contenidor (tenant), usuari i rol administratiu per les operacions administratives.
  - a. Crear el contenidor (tenant) **admin**:

```
santi@Controller:~$ keystone tenant-create --name admin --description "Admin Tenant"
+-----+-----+
| Property | Value |
+-----+-----+
| description | Admin Tenant |
| enabled | True |
| id | 5a927fdf1ef24266b6c9bf555cfc5acc |
| name | admin |
+-----+-----+
```

- b. Crear l'usuari **admin** (password: **adminKeystone**):

```
santi@Controller:~$ keystone user-create --name admin --pass adminKeystone --email admin@example.com
+-----+-----+
| Property | Value |
+-----+-----+
| email | admin@example.com |
| enabled | True |
| id | ee3eb8869a1a44738e166b23f3241cf6 |
| name | admin |
| username | admin |
+-----+-----+
```

- c. Crear el rol **admin** ([per definició](#)<sup>3</sup> de l'OpenStack, és amb permisos administratius):

```
santi@Controller:~$ keystone role-create --name admin
+-----+-----+
| Property | Value |
+-----+-----+
| id | 8763bd2ba80d444d9829b3fecfd6d0f5 |
| name | admin |
+-----+-----+
```

- d. Afegir el rol i l'usuari al contenidor (*tenant*) **admin**:

```
santi@Controller:~$ keystone user-role-add --user admin --tenant admin --role admin
```

---

<sup>3</sup> [http://docs.openstack.org/openstack-ops/content/projects\\_users.html#create\\_new\\_users](http://docs.openstack.org/openstack-ops/content/projects_users.html#create_new_users)

2. Crear un contenidor i un usuari **demo**.
  - a. Crear el contenidor (*tenant*) **demo** (No s'ha de repetir aquest pas quan s'afegeixin usuaris nous per aquest tenant):

```
santi@Controller:~$ keystone tenant-create --name demo --description "Demo tenant"
```

Property	Value
description	Demo tenant
enabled	True
id	55f7067fec4841b09039d7c3589855d7
name	demo

- b. Crear l'usuari i afegir-lo directament al contenidor (tenant) **demo** (Fent servir el paràmetre `--tenant` automàticament s'assigna el rol **member** a l'usuari (sense drets administratius [per definició](#)<sup>4</sup>):

```
santi@Controller:~$ keystone user-create --name demo --tenant demo --pass demoKeystone --email demo@example.com
```

Property	Value
email	demo@example.com
enabled	True
id	f4de368fd8e34727856ca8f22de91ded
name	demo
tenantId	55f7067fec4841b09039d7c3589855d7
username	demo

Els dos passos anteriors serveixen per anar afegint els diferents contenidors (*tenants*) i usuaris que es desitgin.

3. Els serveis de l'OpenStack necessiten un contenidor (*tenant*), un usuari i un rol per interactuar amb altres serveis. Normalment cada servei requereix la creació d'un o varis usuaris (únics) amb el rol **admin** inclosos al contenidor **service**. Així, cal crear aquest contenidor:

```
santi@Controller:~$ keystone tenant-create --name service --description "Service Tenant"
```

Property	Value
description	Service Tenant
enabled	True
id	f31a496ea8e44b478333fe62a1749982
name	service

## Creació de l'entitat de servei i de l'API

### Prerequisites

Les variables d'entorn temporals `OS_SERVICE_TOKEN` i `OS_SERVICE_ENDPOINT` han d'estar configurades correctament, per poder crear contenidors, usuaris i rols:

- `OS_SERVICE_TOKEN = 46b2e7eeacc97e3aac9d`
- `OS_SERVICE_ENDPOINT = http://controller:35357/v2.0`

<sup>4</sup> [http://docs.openstack.org/openstack-ops/content/projects\\_users.html#create\\_new\\_users](http://docs.openstack.org/openstack-ops/content/projects_users.html#create_new_users)

## Creació

1. El servei **Identity (keystone)** controla un catàleg de serveis de l'entorn OpenStack. Els serveis fan servir aquest catàleg per localitzar altres serveis que estan corrent a l'entorn. Diguéssim que és un *catàleg de localització*.

Creació del servei **Identity (keystone)** com a tipus *identity*:

```
santi@Controller:~$ keystone service-create --name keystone --type identity --description "OpenStack Identity"
```

Property	Value
description	OpenStack Identity
enabled	True
id	38f7741f35074c56ab556a1b90c03a91
name	keystone
type	identity

2. A més a més, **Identity (keystone)** també controla un catàleg d'APIs associades amb els serveis de l'entorn OpenStack. Els serveis fan servir aquest catàleg per saber com s'han de comunicar amb els altres serveis de l'entorn.

L'OpenStack proveeix tres variants d'APIs (cadascuna amb el seu nivell de privilegis) per cada servei: **admin**, **internal** i **public**. En un entorn de producció, cada API hauria de residir en una xarxa diferent per donar servei a diferents tipus d'usuaris per raons de seguretat. A més a més, per fer-ho escalable, l'OpenStack dóna suport per funcionar amb múltiples regions. Tot i això, en aquest muntatge, per simplificar les coses, la configuració fa servir la xarxa **management** per totes les variants i una sola regió anomenada **regionOne**.

Creació dels punts d'accés a les diferents variants d'APIs (*public*, *internal*, *admin*), es pot veure que es tracta d'URLs amb el port corresponent, i de la regió:

```
santi@Controller:~$ keystone endpoint-create \  
> --service-id $(keystone service-list | awk '/ identity / {print $2}') \  
> --publicurl http://controller:5000/v2.0 \  
> --internalurl http://controller:5000/v2.0 \  
> --adminurl http://controller:35357/v2.0 \  
> --region regionOne
```

Property	Value
adminurl	http://controller:35357/v2.0
id	cad8e4273bd4615b61b83a1fc615df1
internalurl	http://controller:5000/v2.0
publicurl	http://controller:5000/v2.0
region	regionOne
service_id	38f7741f35074c56ab556a1b90c03a91

Observacions:



- El `service_id` que apareix a la darrera comanda, és el mateix que el que s'ha generat a l'anterior.
- Cada servei que s'afegeixi a l'OpenStack requereix afegir informació per l'**Identity** dels punts d'entrada a l'API d'aquest.

## Verificacions

Verificarem que el servei **Identity** funciona com ho hauria de fer.

1. Desfer les variables d'entorn temporals abans configurades:

```
santi@Controller:~$ unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
```

2. Com a usuari **admin** del contenidor **admin**, sol·licitar un *token* d'autenticació:

```
santi@Controller:~$ keystone --os-tenant-name admin --os-username admin \
> --os-password adminkeystone --os-auth-url http://controller:35357/v2.0 \
> token-get
```

Property	Value
expires	2015-02-24T11:30:38Z
id	46672f1abf614f79b6faf483fdbf20be
tenant_id	5a927fdf1ef24266b6c9bf555cfc5acc
user_id	ee3eb8869a1a44738e166b23f3241cf6

3. Com a usuari **admin** del contenidor **admin**, llistar els contenidors per verificar que el contenidor **admin** i l'usuari poden executar només comandes administratives i que l'**Identity** conté els contenidors que s'han anat creat (amb els seus corresponents ids):

```
santi@Controller:~$ keystone --os-tenant-name admin --os-username admin \
> --os-password adminkeystone --os-auth-url http://controller:35357/v2.0 \
> tenant-list
```

id	name	enabled
5a927fdf1ef24266b6c9bf555cfc5acc	admin	True
55f7067fec4841b09039d7c3589855d7	demo	True
f31a496ea8e44b478333fe62a1749982	service	True

4. Com a usuari **admin** del contenidor **admin**, llistar els usuaris que s'han anat creat:

```
santi@Controller:~$ keystone --os-tenant-name admin --os-username admin \
> --os-password adminkeystone --os-auth-url http://controller:35357/v2.0 \
> user-list
```

id	name	enabled	email
ee3eb8869a1a44738e166b23f3241cf6	admin	True	admin@example.com
f4de368fd8e34727856ca8f22de91ded	demo	True	demo@example.com

5. Com a usuari **admin** del contenidor **admin**, llistar els rols que s'han anat creant:

```
santi@Controller:~$ keystone --os-tenant-name admin --os-username admin \  
> --os-password adminkeystone --os-auth-url http://controller:35357/v2.0 \  
> role-list  
+-----+  
|          id          |      name      |  
+-----+  
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_ |  
| 8763bd2ba80d444d9829b3fecfd6d0f5 | admin     |  
+-----+
```

6. Com a usuari **demo** del contenidor **demo**, sol·licitar un *token* d'autenticació:

```
santi@Controller:~$ keystone --os-tenant-name demo --os-username demo \  
> --os-password demokeystone --os-auth-url http://controller:35357/v2.0 \  
> token-get  
+-----+  
| Property |      Value      |  
+-----+  
| expires  | 2015-02-24T11:48:14Z |  
| id       | 24d3ef8cc7b2452aa2708260bfaeb15a |  
| tenant_id | 55f7067fec4841b09039d7c3589855d7 |  
| user_id  | fade368fd8e34727856ca8f22de91ded |  
+-----+
```

7. Com a usuari **demo** del contenidor **demo**, intentar llistar els usuaris per verificar que no es poden executar comandes administratives:

```
santi@Controller:~$ keystone --os-tenant-name demo --os-username demo \  
> --os-password demokeystone --os-auth-url http://controller:35357/v2.0 \  
> user-list  
You are not authorized to perform the requested action: admin_required (HTTP 403)
```

(\*) Cada servei de l'OpenStack referencia a un fitxer *policy.json* per determinar les operacions disponibles per un contenidor, usuari o rol concrets.

### Creació dels *scripts* de l'entorn client de l'OpenStack

Com s'ha vist a l'apartat de verificacions, per cada comanda *keystone* es necessita passar per paràmetre el contenidor, l'usuari, el password i la url d'accés a l'API corresponent. Per agilitzar aquest procés d'autenticació, es poden definir variables d'entorn temporals (com a l'apartat de creació dels contenidors, usuaris i rols) per agilitzar les operacions *keystone*.

- Credencials d'administració i ubicació dels serveis. Crear el fitxer *admin-openrc.sh*:

```
santi@Controller:~$ vim /home/santi/admin-openrc.sh
```

Afegir el següent contingut:

```
export OS_TENANT_NAME=admin  
export OS_USERNAME=admin  
export OS_PASSWORD=adminkeystone  
export OS_AUTH_URL=http://controller:35357/v2.0
```



- Credencials normals i ubicació dels serveis. Crear el fitxer *demo-openrc.sh*:

```
santi@Controller:~$ cp /home/santi/admin-openrc.sh /home/santi/demo-openrc.sh
santi@Controller:~$ vim demo-openrc.sh
```

Afegir el següent contingut:

```
export OS_TENANT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=demoKeystone
export OS_AUTH_URL=http://controller:5000/v2.0
```

- Per carregar les dades emmagatzemades als *scripts* (entorn de treball pel client) per poder executar les comandes adientment cal fer:

```
santi@Controller:~$ source admin-openrc.sh
```

Després d'haver configurat el *Keystone*, el mapa lògic de serveis implicats és el següent:

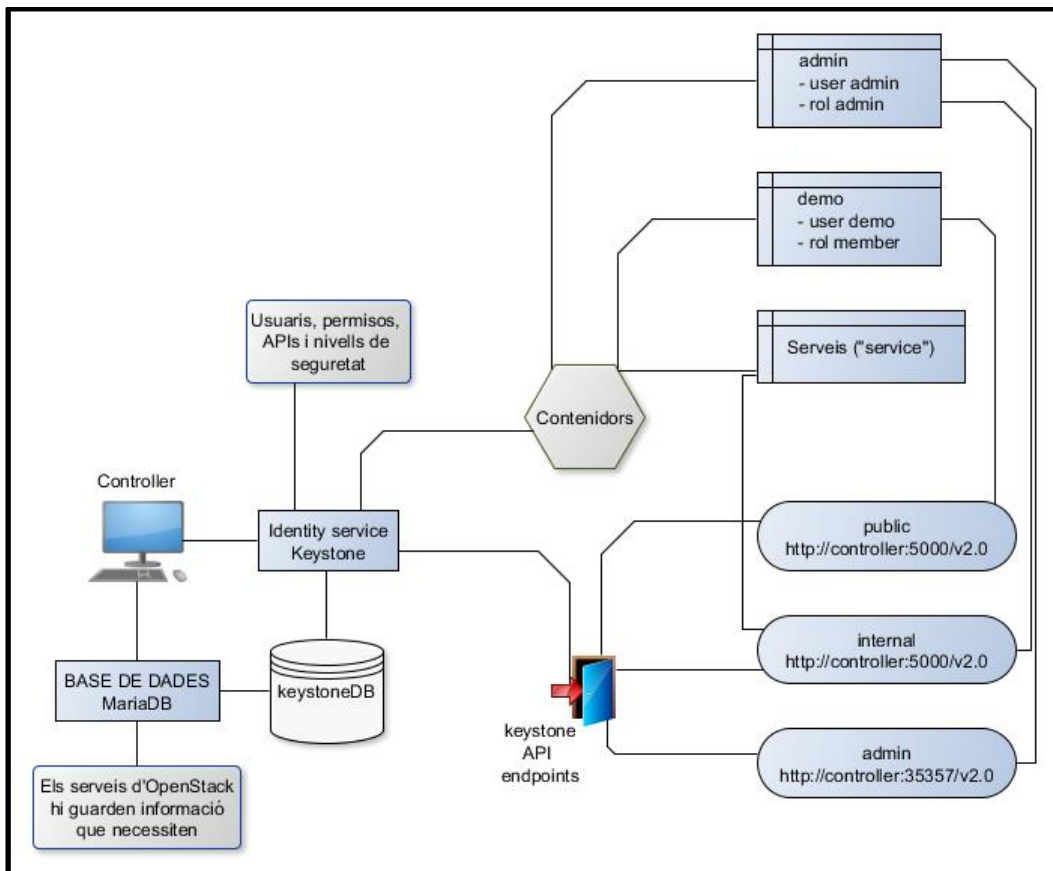


Figura 46: Components Keystone

I el que portem fins ara:

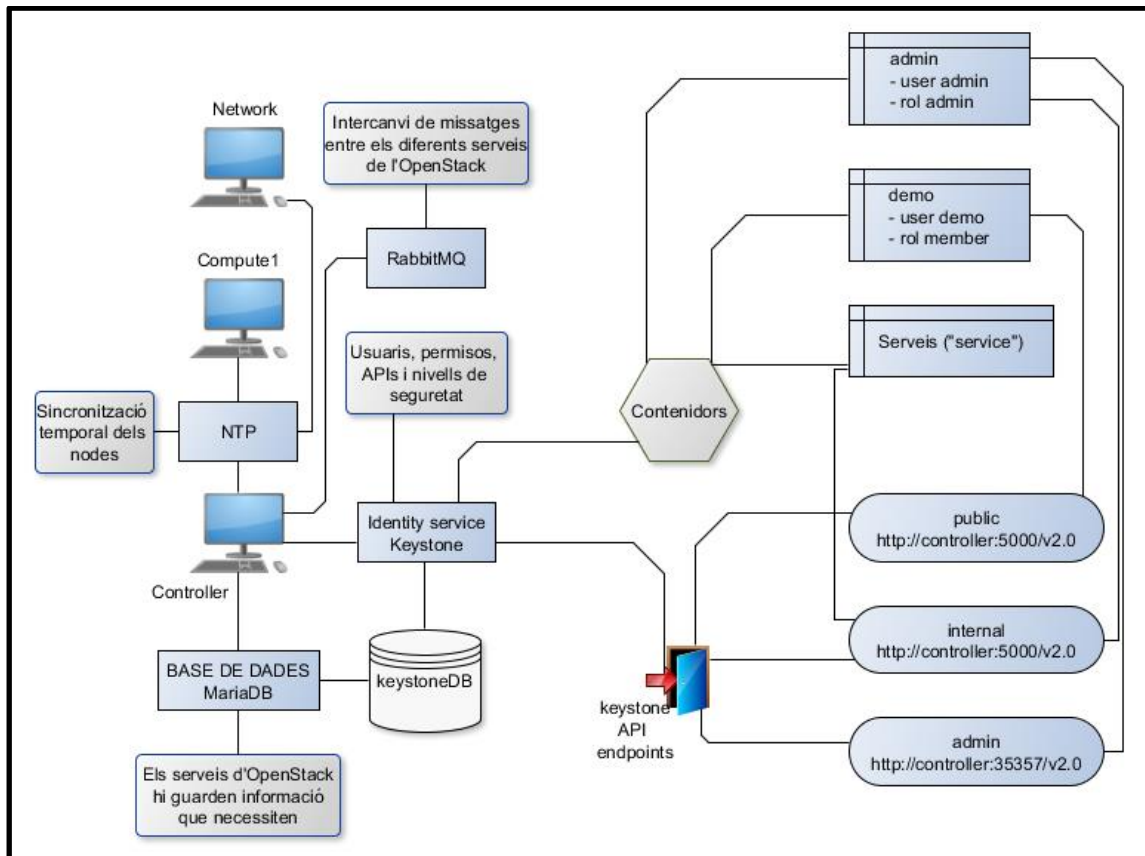


Figura 47: Total de components que es porten fins el moment

### 9.1.2.2 Image service (Glance)

#### Instal·lació i configuració

El servei d'imatges **Glance** s'instal·larà al node **Controller** i guardarà les imatges al sistema de fitxers local.

#### Prerequisits

Abans de realitzar la instal·lació del software pel servei, cal crear la base de dades, les credencials del servei i els punts d'entrada a l'API.

1. Creació de la base de dades. Passos a seguir:
  - a) Accedir a la base de dades com a *root*:

```
santi@Controller:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 74
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

- b) Crear la base de dades **glance**:

```
MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.00 sec)
```

- c) Posar els permisos d'accés a la base de dades per l'usuari *glance*:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
-> IDENTIFIED BY '20glance15';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
-> IDENTIFIED BY '20glance15';
Query OK, 0 rows affected (0.00 sec)
```

2. Carregar les variables d'entorn d'admin pel **keystone** per poder fer comandes d'administració (incloses a l'script `/home/santi/admin-openrc.sh` que abans he editat):

```
santi@Controller:~$ source admin-openrc.sh
```

3. Crear les credencials pel servei **glance**:
  - a) Crear l'usuari **glance**:

```
santi@Controller:~$ keystone user-create --name glance --pass 20glance15
```

Property	Value
email	
enabled	True
id	7c5a9684ce6441a7b87c11bee7d043f2
name	glance
username	glance

- b) Afegir el rol *admin* a l'usuari *glance* del contenidor de serveis:

```
santi@Controller:~$ keystone user-role-add --user glance --tenant service --role admin
```

- c) Crear l'entitat de servei **glance**:

```
santi@Controller:~$ keystone service-create --name glance --type image \
> --description "OpenStack Image Service"
```

Property	Value
description	OpenStack Image Service
enabled	True
id	8c3ac22ff6364c0980f5129ed3068076
name	glance
type	image

4. Creació dels punts d'accés a les diferents variants d'APIs (*public*, *internal*, *admin*), es pot veure que es tracta d'urls amb el port corresponent, i de la regió:

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ image / {print $2}') \
> --publicurl http://controller:9292 \
> --internalurl http://controller:9292 \
> --adminurl http://controller:9292 \
> --region regionOne
```

Property	Value
adminurl	http://controller:9292
id	614ade61abeb46b3bda05e5a2d8d35f3
internalurl	http://controller:9292
publicurl	http://controller:9292
region	regionOne
service_id	8c3ac22ff6364c0980f5129ed3068076

## Instal·lar i configurar els components del servei

1. Instal·lar els paquets:

```
santi@Controller:~$ sudo apt-get install glance python-glanceclient
```

2. Editar el fitxer `/etc/glance/glance-api.conf` amb el següent contingut:
  - a) A la secció `[database]` configurar l'accés a la base de dades:

```
[database]
# The file name to use with SQLite (string value)
sqlite_db = /var/lib/glance/glance.sqlite

# If True, SQLite uses synchronous mode (boolean value)
#sqlite_synchronous = True

# The backend to use for db (string value)
# Deprecated group/name - [DEFAULT]/db_backend
backend = sqlalchemy

# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql://glance:20glance15@controller/glance
```

- b) A les seccions [keystone\_authtoken] i [paste\_deploy], configurar l'accés al servei **Identity** (keystone):

```
[keystone_authtoken]
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = glance
admin_password = 20glance15
revocation_cache_time = 10

[paste_deploy]
# Name of the paste configuration file that defines the available pipelines
#config_file = glance-api-paste.ini

# Partial name of a pipeline in your paste configuration file with the
# service name removed. For example, if your paste section name is
# [pipeline:glance-api-keystone], you would configure the flavor below
# as 'keystone'.
flavor = keystone
```

- c) A la secció [glance\_store], configurar el sistema de fitxers local i el *path* a les imatges:

```
[glance_store]
# List of which store classes and store class locations are
# currently known to glance at startup.
# Existing but disabled stores:
#     glance.store.rbd.Store,
#     glance.store.s3.Store,
#     glance.store.swift.Store,
#     glance.store.sheepdog.Store,
#     glance.store.cinder.Store,
#     glance.store.gridfs.Store,
#     glance.store.vmware_datastore.Store,
#stores = glance.store.filesystem.Store,
#     glance.store.http.Store
default_store = file

# ===== Filesystem Store Options =====

# Directory that the Filesystem backend store
# writes image data to
filesystem_store_datadir = /var/lib/glance/images/
```

- d) A la secció [DEFAULT], configurar el *driver* de notificació a *noop* per inhabilitar les notificacions, ja que aquestes només calen si s'instal·la el servei de **Telemetry**.

```
# Driver or drivers to handle sending notifications. Set to
# 'messaging' to send notifications to a message queue.
notification_driver = noop
```

- e) A la secció [DEFAULT], habilitar el *verbose* per poder fer *logs* del que està passant amb aquest servei:

```
[DEFAULT]
# Show more verbose log output (sets INFO log level output)
verbose = True
```

3. Editar el fitxer */etc/glance/glance-registry.conf* amb el següent contingut:

- a) A la secció [database], configurar l'accés a la base de dades:

```
# The SQLAlchemy connection string used to connect to the
# database (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql://glance:20glance15@controller/glance
```

- b) A les seccions [keystone\_authtoken] i [paste\_deploy], configurar l'accés al servei **Identity** (keystone):

```
[keystone_authtoken]
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = glance
admin_password = 20glance15

[paste_deploy]
# Name of the paste configuration file that defines the available pipelines
#config_file = glance-registry-paste.ini

# Partial name of a pipeline in your paste configuration file with the
# service name removed. For example, if your paste section name is
# [pipeline:glance-registry-keystone], you would configure the flavor below
# as 'keystone'.
flavor = keystone
```

- c) A la secció [DEFAULT], configurar el *driver* de notificació a *noop* per inhabilitar les notificacions, ja que aquestes només calen si s'instal·la el servei de **Telemetry**.

```
# ===== Notification System Options =====
# Driver or drivers to handle sending notifications. Set to
# 'messaging' to send notifications to a message queue.
notification_driver = noop
```

- d) A la secció [DEFAULT], habilitar el *verbose* per poder fer *logs* del que està passant amb aquest servei:

```
[DEFAULT]
# Show more verbose log output (sets INFO log level output)
verbose = True
```

4. Emplenar la base de dades del servei **Image (glance)**:

```
santi@Controller:~$ sudo su -s /bin/sh -c "glance-manage db_sync" glance
```

### Finalització de la instal·lació

1. Reiniciar els serveis **Image (glance)**:

```
santi@Controller:~$ sudo service glance-registry restart
glance-registry stop/waiting
glance-registry start/running, process 31767
santi@Controller:~$ sudo service glance-api restart
glance-api stop/waiting
glance-api start/running, process 31787
```

2. Eliminar la base de dades SQLite que els paquets Ubuntu han creat per defecte, ja que s'està treballant amb SQL server:

```
santi@Controller:~$ sudo rm -f /var/lib/glance/glance.sqlite
```

### Comprovacions

Per comprovar que el servei s'ha instal·lat i configurat correctament, caldrà fer servir una petita imatge d'un Linux anomenada CirrOS. Per saber com baixar i fer imatges mirar [aquí](#)<sup>5</sup> i per informació sobre gestió d'imatges, [aquí](#)<sup>6</sup>.

1. Crear un directori temporal i anar-hi:

```
santi@Controller:~$ mkdir /tmp/images
santi@Controller:~$ cd /tmp/images
```

2. Descarregar la imatge al directori:

```
santi@Controller:/tmp/images$ wget http://cdn.download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img
--2015-02-25 15:25:41-- http://cdn.download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img
Resolving cdn.download.cirros-cloud.net (cdn.download.cirros-cloud.net)... 92.122.214.51, 92.122.214.11, 2a02:26f0:64::170e:5c6b, ...
Connecting to cdn.download.cirros-cloud.net (cdn.download.cirros-cloud.net)|92.122.214.51|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13200896 (13M) [application/octet-stream]
Saving to: 'cirros-0.3.3-x86_64-disk.img'

100%[=====>] 13,200,896 5.30MB/s in 2.4s

2015-02-25 15:25:43 (5.30 MB/s) - 'cirros-0.3.3-x86_64-disk.img' saved [13200896/13200896]
```

3. Posar les credencials d'*admin* pel **keystone**:

---

<sup>5</sup> <http://docs.openstack.org/image-guide/content/index.html>

<sup>6</sup> <http://docs.openstack.org/user-guide/content/index.html>



```
santi@Controller:/tmp/images$ source /home/santi/admin-openrc.sh
```

4. Pujar la imatge al servei **Image (glance)**:

```
santi@Controller:/tmp/images$ glance image-create --name "cirros-0.3.3-x86_64" \  
> --file cirros-0.3.3-x86_64-disk.img \  
> --disk-format qcow2 \  
> --container-format bare \  
> --is-public True \  
> --progress  
[=====>] 100%  
+-----+  
| Property      | Value  
+-----+  
| checksum      | 133eae9fb1c98f45894a4e60d8736619  
| container_format | bare  
| created_at    | 2015-02-25T14:34:19  
| deleted       | False  
| deleted_at    | None  
| disk_format   | qcow2  
| id            | 7585c3a8-cbdb-4c9b-b819-f819331f95e9  
| is_public     | True  
| min_disk      | 0  
| min_ram       | 0  
| name          | cirros-0.3.3-x86_64  
| owner         | 5a927fdf1ef24266b6c9bf555cfc5acc  
| protected     | False  
| size          | 13200896  
| status        | active  
| updated_at    | 2015-02-25T14:34:19  
| virtual_size  | None  
+-----+
```

Per informació sobre els paràmetres de la comanda **glance image-create**, [aquí](#)<sup>7</sup>.

Per informació sobre formats de discos i containers per les imatges, [aquí](#)<sup>8</sup>.

5. Confirmar que s'ha pujat la imatge i que els atributs són els correctes:

```
santi@Controller:/tmp/images$ glance image-list  
+-----+  
| ID              | Name                | Disk Format | Container Format | Size  
| Status         |  
+-----+  
| 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | cirros-0.3.3-x86_64 | qcow2      | bare            | 1320  
0896 | active |  
+-----+
```

6. Eliminar el directori temporal:

```
santi@Controller:~$ rm -r /tmp/images/
```

Després d'haver configurat el *Glance*, el mapa lògic de serveis implicats és el següent:

---

<sup>7</sup> [http://docs.openstack.org/cli-reference/content/glanceclient\\_commands.html#glanceclient\\_subcommand\\_image-create](http://docs.openstack.org/cli-reference/content/glanceclient_commands.html#glanceclient_subcommand_image-create)

<sup>8</sup> <http://docs.openstack.org/image-guide/content/image-formats.html>



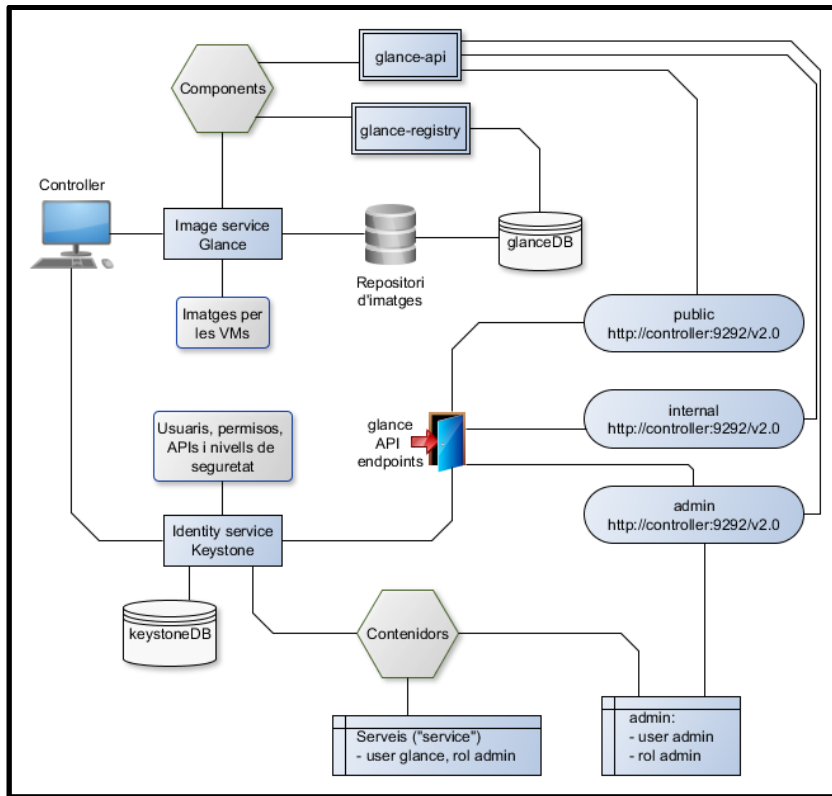


Figura 48: Components Glance

I el que portem fins ara:

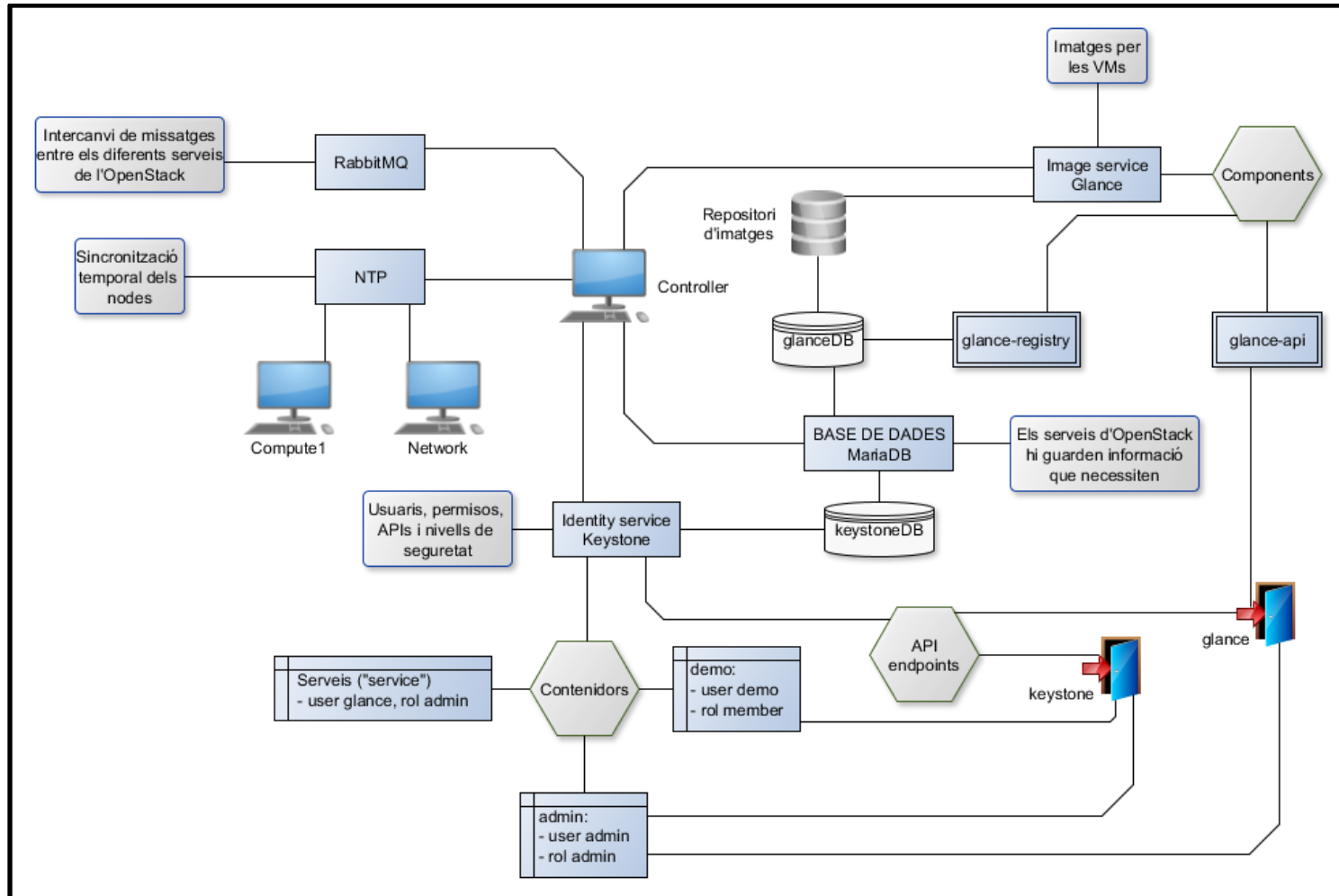


Figura 49: Components totals

### 9.1.2.3 Compute service (Nova)

#### Instal·lació i configuració del node Controller

##### Prerequisites

S'han de realitzar les següents configuracions com a prerequisites:

1. Crear la base de dades pel servei nova:
  - a) Connectar-se amb credencials de *root* a la base de dades:

```
santi@Controller:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 212
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

- b) Crear la base de dades pel servei de computació *nova*:

```
MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.00 sec)
```

- c) Generar les credencials d'accés adequades per la base de dades:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' \
-> IDENTIFIED BY '20nova15';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' \
-> IDENTIFIED BY '20nova15';
Query OK, 0 rows affected (0.00 sec)
```

- a) Crear l'usuari *nova*:

```
santi@Controller:~$ keystone user-create --name nova --pass 20nova15
+-----+-----+
| Property | Value |
+-----+-----+
| email    |       |
| enabled  | True  |
| id       | 81a3151e2c9b4b5baac98d739e3c9b0a |
| name     | nova  |
| username | nova  |
+-----+-----+
```

- b) Afegir el rol *admin* del *tenant service* a l'usuari *nova*:

```
santi@Controller:~$ keystone user-role-add --user nova --tenant service --role admin
```

- c) Crear l'entitat de servei *nova*:

```
santi@Controller:~$ keystone user-role-add --user nova --tenant service --role admin
santi@Controller:~$ keystone service-create --name nova --type compute \
> --description "OpenStack Compute"
+-----+
| Property | Value |
+-----+
| description | OpenStack Compute |
| enabled | True |
| id | 32ab1d84712f4c969a3c774960f9ac01 |
| name | nova |
| type | compute |
+-----+
```

4. Crear les URLs d'entrada (administrativa, pública, interna) a l'API del servei:

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ compute / {print $2}') \
> --publicurl http://controller:8774/v2/^(tenant_id)s \
> --internalurl http://controller:8774/v2/^(tenant_id)s \
> --adminurl http://controller:8774/v2/^(tenant_id)s \
> --region regionOne
+-----+
| Property | Value |
+-----+
| adminurl | http://controller:8774/v2/^(tenant_id)s |
| id | 3dfc50e0c22e41ecbae213e0912401b2 |
| internalurl | http://controller:8774/v2/^(tenant_id)s |
| publicurl | http://controller:8774/v2/^(tenant_id)s |
| region | regionOne |
| service_id | 32ab1d84712f4c969a3c774960f9ac01 |
+-----+
```

### **Instal·lació i configuració del software específic del servei nova**

Un cop configurada la part d'autenticació amb el **keystone**, passaré a configurar els components de software necessaris al node **Controller**.

1. Instal·lar els paquets necessaris:

```
santi@Controller:~$ sudo apt-get install nova-api nova-cert nova-conductor \
> nova-consoleauth nova-novncproxy nova-scheduler python-novaclient_
```

2. Editar el fitxer `/etc/nova/nova.conf`. L'esquelet d'aquest fitxer de configuració no és ni molt menys tan ben estructurat ni comentat com els altres dos que he hagut de tocar anteriorment (`keystone.conf` i `glance-api.conf`), falten seccions a les que es fa referència i no hi ha res comentat. Cap problema, s'afegeixen i ja està, tal i com comentaven en aquest [post<sup>9</sup>](https://ask.openstack.org/en/question/53546/novaconf-has-no-database-section-default/) del fòrum oficial d'OpenStack.

- a) A la secció `[database]`, configurar l'accés a la base de dades:

```
[database]
#
# Database access
#
connection = mysql://nova:20nova15@controller/nova
```

<sup>9</sup> <https://ask.openstack.org/en/question/53546/novaconf-has-no-database-section-default/>

- b) A la secció [DEFAULT], configurar l'accés al servei de missatgeria *RabbitMQ*:

```
#
# RabbitMQ access configuration
#
rpc_backend = rabbit
rabbit_host = controller
rabbit_password = 20openstack15
```

- c) A les seccions [DEFAULT] i [keystone\_authtoken] configurar l'accés al servei **Identity (keystone)**:

```
#
# Authentication
#
auth_strategy = keystone
```

```
[keystone_authtoken]
#
# keystone service access
#
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = nova
admin_password = 20nova15
```

- d) A la secció [DEFAULT], configurar l'opció **my\_ip** per fer servir la interfície *management* del node **Controller**:

```
#
# Use of the management interface IP address
#
my_ip = 10.0.0.11
```

- e) A la secció [DEFAULT], configurar el proxy VNC perquè faci servir la interfície *management* també:

```
#
# VNC proxy configuration
#
vncserver_listen = 10.0.0.11
vncserver_proxyclient_address = 10.0.0.11
```

- f) A la secció [glance], configurar la ubicació del servei **Image (glance)**:

```
[glance]
#
# Image service (glance) location
#
host = controller
```

- g) Habilitar el *verbose* a la secció [DEFAULT]:

```
#
# Verbose logging
#
verbose=True
```

3. Escriure els canvis a la base de dades **nova**:

```
santi@Controller:~$ sudo su -s /bin/sh -c "nova-manage db sync" nova
```

### **Finalitzar la instal·lació**

1. Reiniciar els serveis:

```
santi@Controller:~$ sudo service nova-api restart
nova-api stop/waiting
nova-api start/running, process 24107
santi@Controller:~$ sudo service nova-cert restart
nova-cert stop/waiting
nova-cert start/running, process 24141
santi@Controller:~$ sudo service nova-consoleauth restart
nova-consoleauth stop/waiting
nova-consoleauth start/running, process 24164
santi@Controller:~$ sudo service nova-scheduler restart
nova-scheduler stop/waiting
nova-scheduler start/running, process 24189
santi@Controller:~$ sudo service nova-conductor restart
nova-conductor stop/waiting
nova-conductor start/running, process 24212
santi@Controller:~$ sudo service nova-novncproxy restart
nova-novncproxy stop/waiting
nova-novncproxy start/running, process 24240
```

2. Eliminar la base de dades SQLite que es crea per defecte:

```
santi@Controller:~$ sudo rm -f /var/lib/nova/nova.sqlite
```

### **Instal·lació i configuració d'un node de Computació**

El darrer pas d'aquest apartat és instal·lar i configurar el servei **nova** als nodes de Computació (en aquest cas, i de moment, només serà un). El servei dona suport per diferents *hypervisors* per implementar les instàncies (VMs). En aquest cas, es farà servir l'*hypervisor* QEMU amb l'extensió KVM als nodes que disposin d'acceleració hardware per les màquines virtuals. Pels nodes que ofereixin aquesta característica, es farà servir l'*hypervisor* QEMU genèric.

Per cada node de computació addicional que es vulgui afegir a l'arquitectura, cal:

- Preparar-lo d'una manera semblant al primer, fent servir el mateix servei de xarxa que en aquest.
- Configurar l'NTP (secció d'altres nodes).
- Configurar els *packages* de l'OpenStack.
- Configurar la xarxa seguint l'esquema inicial plantejat (esquema de xarxa).

### **Instal·lació i configuració dels components de l'hypervisor al node Compute1**

1. Instal·lació dels paquets:

```
santi@Compute1:~$ sudo apt-get install nova-compute sysfsutils
```

2. Editar el fitxer de configuració `/etc/nova/nova.conf` amb el següent contingut:
  - a) A la secció `[DEFAULT]`, configurar l'accés al servei de missatgeria *RabbitMQ*:

```
#  
# RabbitMQ access configuration  
#  
rpc_backend = rabbit  
rabbit_host = controller  
rabbit_password = 20openstack15
```

- b) A les seccions `[DEFAULT]` i `[keystone_authtoken]` configurar l'accés al servei **Identity (keystone)**:

```
#  
# Authentication  
#  
auth_strategy = keystone
```

```
[keystone_authtoken]  
#  
# keystone service access  
#  
auth_uri = http://controller:5000/v2.0  
identify_uri = http://controller:35357  
admin_tenant_name = service  
admin_user = nova  
admin_password = 20nova15
```

- c) A la secció `[DEFAULT]`, configurar l'opció **my\_ip** per fer servir la interfície *management*:

```
#  
# Use of the management interface IP address  
#  
my_ip = 10.0.0.31
```

- d) A la secció `[DEFAULT]`, configurar i habilitar l'accés remot a la consola:

```
#  
# Remote console access  
#  
vnc_enabled = True  
vncserver_listen = 0.0.0.0  
vncserver_proxyclient_address = 10.0.0.31  
novncproxy_base_url = http://controller:6080/vnc_auto.html
```

La part servidor escolta a totes les adreces IP i la part *proxy* (consola) només escolta al rang de la interfície de *management* del node de computació. La URL base indica el lloc on es pot fer servir un navegador web per accedir remotament a les consoles de les màquines virtuals d'aquest node de computació. Si el navegador web resideix en un host que no pot resoldre el *hostname controller*, cal canviar `http://controller:6080/vnc_auto.html` per `http://10.0.0.11:6080/vnc_auto.html`.

- e) A la secció [glance], configurar la ubicació del servei **Image (glance)**:

```
[glance]
host = controller
```

- f) Habilitar el *verbose* a la secció [DEFAULT]:

```
#
# Verbose logging
#
verbose=True
```

### Finalització de la instal·lació

1. Determinar si el node de computació té acceleració per hardware per les màquines virtuals:

```
santi@Compute1:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
0
```

Pel resultat que ha donat (0), vol dir que no dóna suport per acceleració hardware. Tot i això, la CPU d'aquest servidor (Intel Xeon E3-1220v3) té VT-x per la virtualització del joc d'instruccions i EPT per a la virtualització de la MMU, característiques necessàries per a l'acceleració hardware de les VMs. Tot i que he habilitat aquestes característiques a posteriori i he intentat carregar el mòdul corresponent, la cosa no funciona.

Com que no tinc suport per acceleració per hardware, hauré de configurar el *libvirt* per fer servir el QEMU enlloc del KVM. Per fer-ho, cal fer un petit canvi a la secció [libvirt] del fitxer de configuració */etc/nova/nova-compute.conf*:

```
[libvirt]
virt_type = qemu
```

2. Reiniciar el servei de Computació (**nova**):

```
santi@Compute1:~$ sudo service nova-compute restart
nova-compute stop/waiting
nova-compute start/running, process 1959
```

3. Eliminar la base de dades SQLite que es crea per defecte:

```
santi@Compute1:~$ sudo rm -f /var/lib/nova/nova.sqlite
```

### Comprovacions

Per realitzar les comprovacions, cal fer-ho des del node Controller.

1. Carregar les credencials d'*admin*:

```
santi@Controller:~$ source admin-openrc.sh
```



2. Llistar els tots els components del servei per verificar que s'estan executant correctament cadascun d'ells:

```
santi@Controller:~$ nova service-list
```

Id	Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
1	nova-cert	Controller	internal	enabled	up	2015-03-02T15:46:19.000000	-
2	nova-consoleauth	Controller	internal	enabled	up	2015-03-02T15:46:21.000000	-
3	nova-scheduler	Controller	internal	enabled	up	2015-03-02T15:46:13.000000	-
4	nova-conductor	Controller	internal	enabled	up	2015-03-02T15:46:13.000000	-
5	nova-compute	Compute1	nova	enabled	up	2015-03-02T15:46:13.000000	-

Com es pot comprovar a la taula, tots els components s'estan executant al node Controller, tret del *nova-compute* que s'està executant al Compute1.

3. Llistar les imatges del catàleg del servei Image (**glance**) per verificar la connectivitat entre el **keystone** i el **glance**:

```
santi@Controller:~$ nova image-list
```

ID	Name	Status	Server
7585c3a8-cbdb-4c9b-b819-f819331f95e9	cirros-0.3.3-x86_64	ACTIVE	

Els components implicats en la configuració que s'ha realitzat pel nova són:

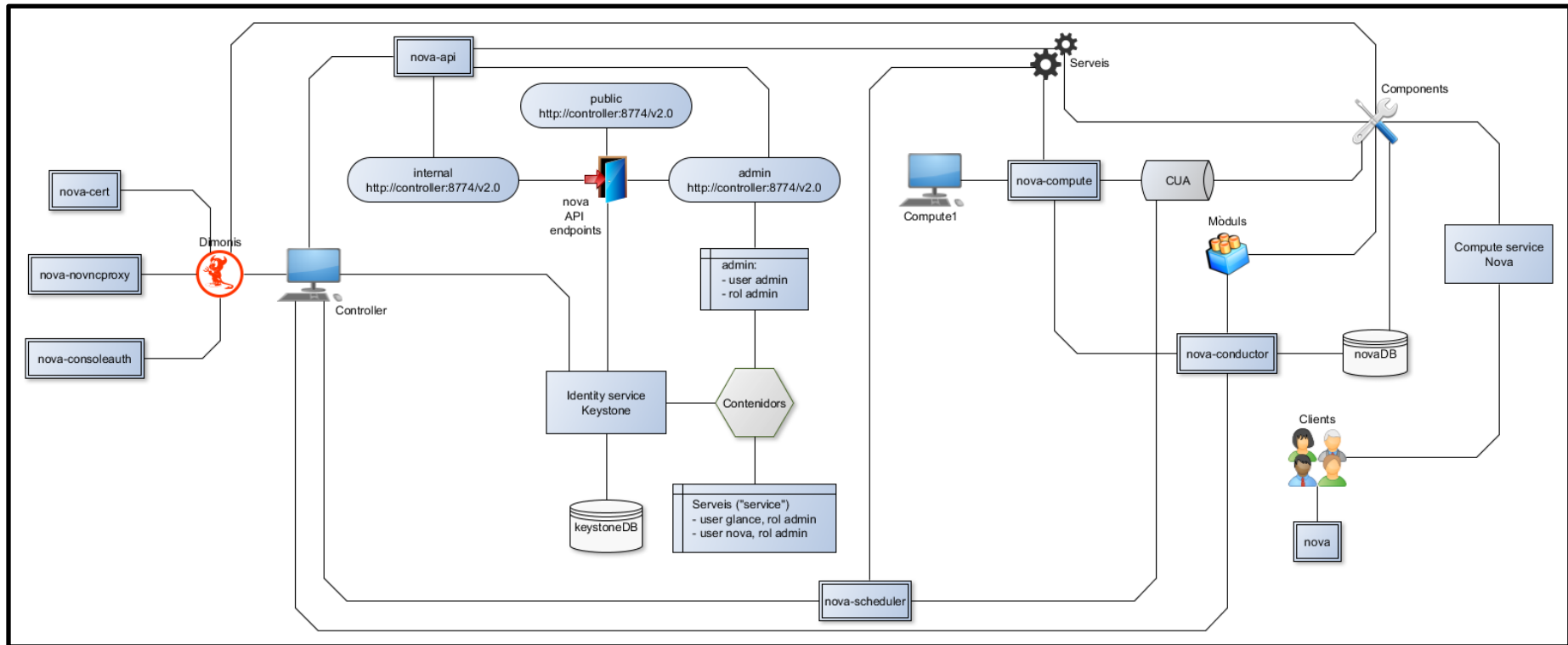


Figura 50: Components del Nova.

## 9.1.2.4 Networking component

### Instal·lació i configuració al node Controller

#### Prerequisites

1. Crear la base de dades:
  - a) Entrar com a root:

```
santi@Controller:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 263
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

- b) Crear la base de dades **neutron**:

```
MariaDB [(none)]> CREATE DATABASE neutron;
Query OK, 1 row affected (0.00 sec)
```

- c) Generar les credencials d'accés adequades per la base de dades:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' \
-> IDENTIFIED BY '20neutron15';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' \
-> IDENTIFIED BY '20neutron15';
Query OK, 0 rows affected (0.00 sec)
```

2. Carregar les credencials per *admin* del **keystone**:

```
santi@Controller:~$ source admin-openrc.sh
```

3. Crear les credencials pel servei al **keystone**:

- a) Crear l'usuari *neutron*:

```
santi@Controller:~$ keystone user-create --name neutron --pass 20neutron15
+-----+-----+
| Property |          Value          |
+-----+-----+
| email    |                          |
| enabled  |             True       |
| id       | 3328e2edffb942b89c3accff397818fa |
| name     |             neutron    |
| username |             neutron    |
+-----+-----+
```

- b) Afegir el rol *admin* del tenant *service* a l'usuari *neutron*:

```
santi@Controller:~$ keystone user-role-add --user neutron --tenant service --role admin
```

- c) Crear l'entitat de servei *neutron*:

```
santi@Controller:~$ keystone service-create --name neutron --type network \
> --description "OpenStack Networking"
+-----+-----+
| Property | Value |
+-----+-----+
| description | OpenStack Networking |
| enabled | True |
| id | 78bff904dfe5493ea59671d12c8d84fa |
| name | neutron |
| type | network |
+-----+-----+
```

- d) Crear les URLs d'entrada (administrativa, pública, interna) a l'API del servei:

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ network / {print $2}') \
> --publicurl http://controller:9696 \
> --adminurl http://controller:9696 \
> --internalurl http://controller:9696 \
> --region regionOne
+-----+-----+
| Property | Value |
+-----+-----+
| adminurl | http://controller:9696 |
| id | fcf249a18bd74d0296812ddcefe43a2e |
| internalurl | http://controller:9696 |
| publicurl | http://controller:9696 |
| region | regionOne |
| service_id | 78bff904dfe5493ea59671d12c8d84fa |
+-----+-----+
```

### **Instal·lació i configuració del software específic del servei neutron**

Un cop configurada la part d'autenticació amb el **keystone**, passaré a configurar els components de software necessaris al node **Controller**.

1. Instal·lar els paquets necessaris:

```
santi@Controller:~$ sudo apt-get install neutron-server neutron-plugin-ml2 python-neutronclient
```

2. Editar el fitxer `/etc/neutron/neutron.conf` amb el següent contingut:
  - a) A la secció `[database]`, configurar l'accés a la base de dades:

```
[database]
# This line MUST be changed to actually run the plugin.
# Example:
connection = mysql://neutron:20neutron15@controller/neutron
```

- b) A la secció `[DEFAULT]`, configurar l'accés al servei de missatgeria *RabbitMQ*:

```
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
rpc_backend=rabbit
```

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c) A les seccions [DEFAULT] i [keystone\_authtoken] configurar l'accés al servei **Identity (keystone)**:

```
# The strategy to be used for auth.
# Supported values are 'keystone' (default), 'noauth'.
auth_strategy = keystone
```

```
[keystone_authtoken]
# auth_host = 127.0.0.1
# auth_port = 35357
# auth_protocol = http
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = neutron
admin_password = 20neutron15
```

- d) A la secció [DEFAULT], habilitar el plugin *Modular Layer 2 (ML2)*, el servei d'encaminament i el solapament d'adreces IP:

```
# (StrOpt) Neutron core plugin endpoint to be loaded from the
# neutron.core_plugins namespace. See setup.cfg for the endpoint names of the
# plugins included in the neutron source distribution. For compatibility with
# previous versions, the class name of a plugin can be specified instead of its
# endpoint name.
#
core_plugin = ml2
```

```
# (ListOpt) List of service plugin endpoints to be loaded from the
# neutron.service_plugins namespace. See setup.cfg for the endpoint names of
# the plugins included in the neutron source distribution. For compatibility
# with previous versions, the class name of a plugin can be specified instead
# of its endpoint name.
#
service_plugins = router
```

```
# Enable or disable overlapping IPs for subnets
# Attention: the following parameter MUST be set to False if Neutron is
# being used in conjunction with nova security groups
allow_overlapping_ips = True
```

- e) A la secció [DEFAULT], configurar el **neutron** perquè notifiqui al **nova** sobre canvis en topologies de xarxes (per obtenir l'id del *tenant service* es pot fer amb la comanda *keystone tenant-get service*, amb les credencials d'*admin*, *source admin-openrc.sh*):

```

# ===== neutron nova interactions =====
# Send notification to nova when port status is active.
notify_nova_on_port_status_changes = True

# Send notifications to nova when port data (fixed_ips/floatingips) change
# so nova can update it's cache.
notify_nova_on_port_data_changes = True

# URL for connection to nova (Only supports one nova region currently).
nova_url = http://controller:8774/v2

# Name of nova region to use. Useful if keystone manages more than one region
nova_region_name = regionOne

# Username for connection to nova in admin context
nova_admin_username = nova

# The uuid of the admin nova tenant
nova_admin_tenant_id = f31a496ea8e44b478333fe62a1749982

# Password for connection to nova in admin context.
nova_admin_password = 20nova15

# Authorization URL for connection to nova in admin context.
nova_admin_auth_url = http://controller:35357/v2.0

```

f) Habilitar el *verbose* a la secció [DEFAULT]:

```

[DEFAULT]
# Print more verbose output (set logging level to INFO instead of default WARNING level).
verbose = True

```

### **Configurar el plugin Modular Layer 2 (ML2)**

Aquest plugin fa servir Open vSwitch (OVS) com a mecanisme per construir xarxes virtuals per les VMs. Tot i això, el node **Controller** no necessita els components de l'OVS perquè aquest no s'encarrega de la gestió de la xarxa de les VMs.

Passos a seguir per la configuració:

1. Editar el fitxer `/etc/neutron/plugins/ml2/ml3_conf.ini` amb el següent contingut:
  - a) A la secció `[ml2]`, habilitar l'encapsulació plana i la genèrica per l'encaminament (*flat and generic routing encapsulation* - GRE) pels tipus de *drivers* de xarxa; el GRE pels tipus de contenidors de xarxa i el *driver* per l'OVS:

```
[ml2]
# (ListOpt) List of network type driver endpoints to be loaded from
# the neutron.ml2.type_drivers namespace.
#
# type_drivers = local,flat,vlan,gre,vxlan
type_drivers = flat,gre
# Example: type_drivers = flat,vlan,gre,vxlan

# (ListOpt) Ordered list of network_types to allocate as tenant
# networks. The default value 'local' is useful for single-box testing
# but provides no connectivity between hosts.
#
# tenant_network_types = local
tenant_network_types = gre
# Example: tenant_network_types = vlan,gre,vxlan

# (ListOpt) Ordered list of networking mechanism driver endpoints
# to be loaded from the neutron.ml2.mechanism_drivers namespace.
mechanism_drivers = openvswitch
```

b) A la secció [ml2\_type\_gre], configurar el rang d'identificadors de túnel:

```
[ml2_type_gre]
# (ListOpt) Comma-separated list of <tun_min>:<tun_max> tuples enumerating ranges of GRE tunnel IDs
# that are available for tenant network allocation
tunnel_id_ranges = 1:1000
```

c) A la secció [securitygroup], habilitar els grups de seguretat; el *ipset* i el *driver* OVS iptables pel firewall:

```
[securitygroup]
# Controls if neutron security group is enabled or not.
# It should be false when you use nova security group.
enable_security_group = True

# Use ipset to speed-up the iptables security groups. Enabling ipset support
# requires that ipset is installed on L2 agent node.
enable_ipset = True

# OVS iptables firewall driver
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

### Configurar el Compute (nova) perquè utilitzi el Networking (neutron)

La configuració per defecte és per al mode *legacy networking*. Com el mode triat és el **neutron**, cal realitzar la configuració correcta.

1. Editar el fitxer de configuració del **nova** (*/etc/nova/nova.conf*) amb el següent contingut:
  - a) A la secció [DEFAULT], configurar les APIs i els *drivers*:



```

#
# API configurations
#
network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

#
# Driver configurations
#
linuxnet_interface_driver = nova.network.linux_net.LinuxOVNInterfaceDriver
# Disable default nova driver, to use neutron's firewall
firewall_driver = nova.virt.firewall.NoopFirewallDriver

```

- b) A la secció [neutron], configurar els paràmetres d'accés:

```

[neutron]
#
# Access parameters
#
url = http://controller:9696
auth_strategy = keystone
admin_auth_url = http://controller:35357/v2.0
admin_tenant_name = service
admin_username = neutron
admin_password = 20neutron15

```

### Finalització de la instal·lació

1. Actualitzar la base de dades *neutron* amb els canvis fets:

```

santi@Controller:~$ sudo su -s /bin/sh -c "neutron-db-manage \
> --config-file /etc/neutron/neutron.conf \
> --config-file /etc/neutron/plugins/ml2/ml2_conf.ini \
> upgrade junos \
> neutron

```

2. Reiniciar els serveis que depenen del Compute (**nova**):

```

nova-api start/running, process 9973
santi@Controller:~$ sudo service nova-scheduler restart
nova-scheduler stop/waiting
nova-scheduler start/running, process 10007
santi@Controller:~$ sudo service nova-conductor restart
nova-conductor stop/waiting
nova-conductor start/running, process 10030

```

3. Reiniciar el servei Networking (neutron):

```

santi@Controller:~$ sudo service neutron-server restart
stop: Unknown instance:
neutron-server start/running, process 10061

```



## Verificacions

1. Carregar les credencials d'*admin*:

```
santi@Controller:~$ source admin-openrc.sh
```

2. Llistar les extensions carregades per verificar que el *neutron-server* s'està executant correctament:

```
santi@Controller:~$ neutron ext-list
+-----+
| alias          | name                               |
+-----+-----+
| security-group | security-group                     |
| l3_agent_scheduler | L3 Agent Scheduler                 |
| ext-gw-mode    | Neutron L3 Configurable external gateway mode |
| binding       | Port Binding                       |
| provider      | Provider Network                   |
| agent        | agent                               |
| quotas       | Quota management support           |
| dhcp_agent_scheduler | DHCP Agent Scheduler               |
| l3-ha        | HA Router extension                |
| multi-provider | Multi Provider Network             |
| external-net  | Neutron external network           |
| router       | Neutron L3 Router                  |
| allowed-address-pairs | Allowed Address Pairs             |
| extraroute    | Neutron Extra Route                |
| extra_dhcp_opt | Neutron Extra DHCP opts            |
| dvr          | Distributed Virtual Router         |
+-----+-----+
```

## Instal·lació i configuració del node Network

El node Network principalment s'encarrega de l'encaminament entre les xarxes internes i l'externa i del DHCP per les xarxes virtuals.

### Prerequisits

1. Habilitar l'*ip\_forwarding* i no realitzar cap comprovació sobre l'*spoofing* de connexions o enviament de paquets per fer DDoS ([font<sup>10</sup>](http://font10)):

```
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

2. Aplicar els canvis:

---

<sup>10</sup> <http://unaaldia.hispasec.com/2002/11/opciones-de-seguridad-en-linux-traves.html>

```
santi@Network:~$ sudo sysctl -p
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.ip_forward = 1
```

### Instal·lació dels paquets específics de Networking (neutron)

```
santi@Network:~$ sudo apt-get install neutron-plugin-m12 \
> neutron-plugin-openvswitch-agent \
> neutron-l3-agent \
> neutron-dhcp-agent
```

### Configuració dels components del Networking (neutron)

La configuració bàsica del **neutron** al node Network consisteix en el mecanisme d'autenticació, la missatgeria i el *plugin*.

1. Editar el fitxer de configuració del **neutron** (*/etc/neutron/neutron.conf*) amb el següent contingut:
  - a) A la secció [database], cal comentar qualsevol línia *connection* perquè els nodes Network no han d'accedir directament a la base de dades.

```
[database]
# This line MUST be changed to actually run the plugin.
# Example:
# connection = mysql://root:pass@127.0.0.1:3306/neutron
# Replace 127.0.0.1 above with the IP address of the database used by the
# main neutron server. (Leave it as is if the database runs on this host.)
# connection = sqlite:///var/lib/neutron/neutron.sqlite
# NOTE: In deployment the [database] section and its connection attribute may
# be set in the corresponding core plugin '.ini' file. However, it is suggested
# to put the [database] section and its connection attribute in this
# configuration file.
```

- b) A la secció [DEFAULT], configurar l'accés al *RabbitMQ*:

```
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
rpc_backend = rabbit
```

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c) A la secció [DEFAULT] i [keystone-authtoken], configurar l'accés al servei **Identity (keystone)**:

```
# The strategy to be used for auth.
# Supported values are 'keystone' (default), 'noauth'.
auth_strategy = keystone
```

```
[keystone_auth_token]
# auth_host = 127.0.0.1
# auth_port = 35357
# auth_protocol = http
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = neutron
admin_password = 20neutron15
```

- d) A la secció [DEFAULT], habilitar el *plugin Modular Layer 2 (ML2)*, el servei de d'encaminament i el solapament d'IPs.

```
# (ListOpt) List of service plugin entrypoints to be loaded from the
# neutron.service_plugins namespace. See setup.cfg for the entrypoint names of
# the plugins included in the neutron source distribution. For compatibility
# with previous versions, the class name of a plugin can be specified instead
# of its entrypoint name.
#
service_plugins = router
# Example: service_plugins = router,firewall,lbaas,vpnaas,metering

# Paste configuration file
# api_paste_config = api-paste.ini

# The strategy to be used for auth.
# Supported values are 'keystone' (default), 'noauth'.
auth_strategy = keystone
```

```
# Enable or disable overlapping IPs for subnets
# Attention: the following parameter MUST be set to False if Neutron is
# being used in conjunction with nova security groups
allow_overlapping_ips = True
```

- e) A la secció [DEFAULT], habilitar el *verbose*:

```
[DEFAULT]
# Print more verbose output (set logging level to INFO instead of default WARNING level).
verbose = True
```

### **Configurar el plugin Modular Layer 2 (ML2)**

1. Completar la següent informació al fitxer `/etc/neutron/plugins/ml2/ml2_conf.ini`:
  - a) A la secció [ml2], habilitar l'encapsulació plana i la genèrica per l'encaminament (*flat and genèric routing encapsulation - GRE*) pels tipus de *drivers* de xarxa; el GRE pels tipus de contenidors de xarxa i el *driver* per l'OVS:

```
[ml2]
# (ListOpt) List of network type driver endpoints to be loaded from
# the neutron.ml2.type_drivers namespace.
#
# type_drivers = local,flat,vlan,gre,vxlan
type_drivers = flat,gre
# Example: type_drivers = flat,vlan,gre,vxlan

# (ListOpt) Ordered list of network_types to allocate as tenant
# networks. The default value 'local' is useful for single-box testing
# but provides no connectivity between hosts.
#
# tenant_network_types = local
tenant_network_types = gre
# Example: tenant_network_types = vlan,gre,vxlan

# (ListOpt) Ordered list of networking mechanism driver endpoints
# to be loaded from the neutron.ml2.mechanism_drivers namespace.
mechanism_drivers = openvswitch
```

- b) A la secció [ml2\_type\_flat], configurar la xarxa externa com a plana (*flat*):

```
[ml2_type_flat]
# (ListOpt) List of physical_network names with which flat networks
# can be created. Use * to allow flat networks with arbitrary
# physical_network names.
#
flat_networks = external
```

- c) A la secció [ml2\_type\_gre], configurar el rang d'identificadors de túnel:

```
[ml2_type_gre]
# (ListOpt) Comma-separated list of <tun_min>:<tun_max> tuples enumerating ranges of GRE tunnel IDs
# that are available for tenant network allocation
tunnel_id_ranges = 1:1000
```

- d) A la secció [securitygroup], habilitar els grups de seguretat; el *ipset* i el driver OVS *iptables* pel *firewall*:

```
[securitygroup]
# Controls if neutron security group is enabled or not.
# It should be false when you use nova security group.
enable_security_group = True

# Use ipset to speed-up the iptables security groups. Enabling ipset support
# requires that ipset is installed on L2 agent node.
enable_ipset = True

# OVS iptables firewall driver
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

- e) A la secció [ovs], habilitar els túnels, configurar el punt d'entrada del túnel local i mapejar la xarxa externa plana (*flat*) cap al bridge *br-ex* extern:



```
[ovs]
# Local instance tunnels network interface IP
local_ip = 10.0.1.21
# Tunnels enabled
enable_tunneling = True
# Map external flat provider network to the br-ex external network bridge
bridge_mappings = external:br-ex
```

f) A la secció [agent], habilitar els túnels GRE:

```
[agent]
tunnel_types = gre
```

### Configurar l'agent Layer-3

Aquest agent el que fa és proveir serveis d'encaminament per xarxes virtuals. Cal editar el fitxer `/etc/neutron/l3_agent.ini` amb la següent informació:

1. A la secció [DEFAULT], configurar el *driver*, l'espai de noms per les xarxes, configurar el *bridge* de la xarxa externa i habilitar l'eliminació de l'espai de noms per *routers* que estiguin "morts":

```
# L3 requires that an interface driver be set. Choose the one that best
# matches your plugin.
# interface_driver =

# Example of interface_driver option for OVS based plugins (OVS, Ryu, NEC)
# that supports L3 agent
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
```

```
# Name of bridge used for external network traffic. This should be set to
# empty value for the linux bridge. when this parameter is set, each L3 agent
# can be associated with no more than one external network.
external_network_bridge = br-ex
```

```
# router_delete_namespaces, which is false by default, can be set to True if
# namespaces can be deleted cleanly on the host running the L3 agent.
# Do not enable this until you understand the problem with the Linux iproute
# utility mentioned in https://bugs.launchpad.net/neutron/+bug/1052535 and
# you are sure that your version of iproute does not suffer from the problem.
# If True, namespaces will be deleted when a router is destroyed.
router_delete_namespaces = True
```

2. Habilitar el *verbose* a la mateixa secció:

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
debug = True
```

### Configurar l'agent DHCP

Aquest agent proveeix serveis DHCP per les xarxes virtuals.

1. Cal editar el fitxer `/etc/neutron/dhcp_agent.ini` amb la següent informació:

- a) A la secció [DEFAULT], configurar els *drivers*, habilitar els espais de noms i habilitar l'eliminació d'espais de noms DHCP que estiguin "morts":

```
# Example of interface_driver option for OVS based plugins(OVS, Ryu, NEC, NUP,
# BigSwitch/Floodlight)
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
```

```
# The agent can use other DHCP drivers. Dnsmasq is the simplest and requires
# no additional setup of the DHCP server.
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
```

```
# Allow overlapping IP (Must have kernel build with CONFIG_NET_NS=y and
# iproute2 package that supports namespaces).
use_namespaces = True
```

```
# dhcp_delete_namespaces, which is false by default, can be set to True if
# namespaces can be deleted cleanly on the host running the dhcp agent.
# Do not enable this until you understand the problem with the Linux iproute
# utility mentioned in https://bugs.launchpad.net/neutron/+bug/1052535 and
# you are sure that your version of iproute does not suffer from the problem.
# If True, namespaces will be deleted when a dhcp server is disabled.
dhcp_delete_namespaces = False
```

- b) Habilitar el *verbose* a la mateixa secció:

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
debug = True
```

2. Aquesta part és opcional, però és molt interessant implementar-la, i per tant, ho faig. Els protocols de *tunneling* com el GRE afegeixen capçaleres als paquets que fan incrementar l'*overhead*, fent que disminueixi l'espai disponible pel *payload* o les dades d'usuari. Sense tenir coneixement de la infraestructura de la xarxa, les instàncies (VMs) intenten enviar paquets fent servir la mida màxima de paquet *Ethernet* (1500 bytes). A les xarxes IP, es fa servir el mecanisme *path MTU discovery* (PMTUD), que ajusta la mida de les MTUs segons els extrems. Tot i això, alguns sistemes operatius i/o xarxes, bloquegen o no donen suport per aquest mecanisme, causant una degradació del rendiment o la pèrdua de la connexió.

Idealment, es poden prevenir aquest problemes habilitant els *jumbo frames* a la xarxa física que contingui les xarxes virtuals. Aquests poden contenir MTUs de fins a 9000 bytes aproximadament, cosa que fa que es pugui ignorar l'impacte de l'*overhead* provocat pel GRE. Tot i això, molts dispositius de xarxa no tenen suport per aquests paquets, fent perdre el control sobre la infraestructura de la xarxa als administradors d'OpenStack. Tenint en compte aquest problema, el que es pot fer és reduir les MTUs de les VMs perquè tinguin en compte l'*overhead* del GRE. Per determinar la mida ideal, cal experimentar, però 1454 bytes sol funcionar en la majoria dels casos. Així, es pot configurar el servidor DHCP perquè assigni adreces IP a les instàncies i que a més ajusti l'MTU. Hi ha algunes imatges per cloud que ignoren l'opció DHCP MTU, en aquest cas s'hauria d'ajustar fent servir metadades, scripts o d'altres mètodes.

- a) Editar el fitxer `/etc/neutron/dhcp_agent.ini` amb la següent informació:
  - i. A la secció [DEFAULT], habilitar el fitxer de configuració `dnsmasq`:

```
# Override the default dnsmasq settings with this file
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf
```

- b) Crear i editar el fitxer `/etc/neutron/dnsmasq-neutron.conf` amb el següent contingut:

```
#Enable the DHCP MTU option (26) and configure it to 1454 bytes
dhcp-option-force=26,1454
```

- c) Eliminar qualsevol procés `dnsmasq`:

```
santi@Network:~$ sudo pkill dnsmasq
```

### Configurar l'agent de metadades

Aquest agent proveeix informació de configuració com ara credencials per les instàncies.

1. Editar el fitxer `/etc/neutron/metadata_agent.ini` amb el següent contingut:
  - a. A la secció [DEFAULT], configurar els paràmetres d'accés a l'Identity (keystone):

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
# debug = True

# The Neutron user information for accessing the Neutron API.
auth_url = http://localhost:5000/v2.0
auth_region = RegionOne
# Turn off verification of the certificate for ssl
# auth_insecure = False
# Certificate Authority public key (CA cert) file for ssl
# auth_ca_cert =
admin_tenant_name = service
admin_user = neutron
admin_password = 20neutron15
```

- b. A la secció [DEFAULT], configurar el host de metadades:

```
# IP address used by Nova metadata server
nova_metadata_ip = controller
```

- c. A la secció [DEFAULT], configurar el secret compartit pel `proxy` de les metadades:

```
# When proxying metadata requests, Neutron signs the Instance-ID header with a
# shared secret to prevent spoofing. You may select any string for a secret,
# but it must match here and in the configuration used by the Nova Metadata
# Server. NOTE: Nova uses a different key: neutron_metadata_proxy_shared_secret
metadata_proxy_shared_secret = 20metadataSecret15
```

d. A la secció [DEFAULT], habilitar el *verbose*:

```
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output)
debug = True
```

2. Al node **Controller**, editar el fitxer `/etc/nova/nova.conf` amb el següent contingut:
  - a. A la secció [neutron], habilitar el *proxy* de metadades i configurar-ne el secret:

```
#
# Metadata configuration
#
service_metadata_proxy = True
metadata_proxy_shared_secret = 20metadataSecret15
```

3. Al node **Controller**, reiniciar el servei que s'encarrega de l'API del **nova (Compute)**:

```
santi@Controller:~$ sudo service nova-api restart
nova-api stop/waiting
nova-api start/running, process 31632
```

### Configurar el servei Open vSwitch (OVS)

Aquest servei proporciona un conjunt d'eines que permeten muntar les xarxes virtuals per les VMs. La integració del *bridge br-int* gestiona el trànsit de xarxa intern de les VMs amb l'OVS. D'altra banda, el *bridge br-ex* s'encarrega de gestionar el trànsit de xarxa extern de les VMs amb l'OVS. El *bridge* extern requereix un port a la interfície de la xarxa externa per connectar les instàncies amb aquesta. Bàsicament, aquest port connecta les xarxes externes virtual i física.

Tasques a desenvolupar:

1. Reiniciar el servei OVS:

```
santi@Network:~$ sudo service openvswitch-switch restart
[sudo] password for santi:
openvswitch-switch stop/waiting
openvswitch-switch start/running
```

2. Afegir el *bridge* extern:

```
santi@Network:~$ sudo ovs-vsctl add-br br-ex
```

3. Afegir un port al *bridge* que s'acaba de crear que connecti amb la interfície de la xarxa física externa (eth2 en aquest cas):

```
santi@Network:~$ sudo ovs-vsctl add-port br-ex eth2
```

Depenent del *driver* de la interfície, cal desactivar el *generic receive offload (GRO)* per aconseguir un rendiment òptim entre les instàncies i la xarxa externa.



Per deshabilitar el GRO temporalment a la interfície de xarxa externa mentre es testeja l'entorn:

```
santi@Network:~$ sudo ethtool -K eth2 gro off
```

### Finalització de la instal·lació

Reiniciar els serveis del **Networking (neutron)**:

```
santi@Network:~$ sudo service neutron-plugin-openvswitch-agent restart
neutron-plugin-openvswitch-agent stop/waiting
neutron-plugin-openvswitch-agent start/running, process 8906
santi@Network:~$ sudo service neutron-l3-agent restart
stop: Unknown instance:
neutron-l3-agent start/running, process 9118
santi@Network:~$ sudo service neutron-dhcp-agent restart
neutron-dhcp-agent stop/waiting
neutron-dhcp-agent start/running, process 9204
santi@Network:~$ sudo service neutron-metadata-agent restart
neutron-metadata-agent stop/waiting
neutron-metadata-agent start/running, process 9244
```

### Verificacions

Per fer-les, cal ser al node **Controller**:

1. Carregar les credencials d'*admin*:

```
santi@Controller:~$ source admin-openrc.sh
```

2. Llistar els agents per verificar que aquests s'estan executant correctament:

```
santi@Controller:~$ neutron agent-list
+-----+-----+-----+-----+-----+-----+
| id                | agent_type      | host      | alive | admin_state_up | binary |
+-----+-----+-----+-----+-----+-----+
| 1127e434-4856-41f9-87f3-589e96e4fe3d | Open vSwitch agent | Network  | :- )  | True            | neutron-openvswitch-agent |
| 4321b218-b759-4fac-940b-6ecdabc4916e9 | Metadata agent    | Network  | :- )  | True            | neutron-metadata-agent    |
| 9ecb5ba2-7d07-439b-a8e6-076332d362b5 | DHCP agent        | Network  | :- )  | True            | neutron-dhcp-agent        |
| c491468f-f5a3-42b6-845c-d3867c2aa77d | L3 agent          | Network  | :- )  | True            | neutron-l3-agent          |
+-----+-----+-----+-----+-----+-----+

```

### Instal·lació i configuració del node Compute1

Aquest node s'encarrega de la connectivitat i de la seguretat dels grups (*security groups*) per les instàncies.

#### Prerequisits

Els prerequisits fan referència a 2 paràmetres del *kernel* que s'han de retocar, passos a seguir:

1. Editar el fitxer */etc/sysctl.conf* amb el següent contingut:

```
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0
```

2. Aplicar els canvis:

```
santi@Compute1:~$ sudo sysctl -p
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
```

### **Instal·lació dels components del Networking (neutron)**

Instal·lar els paquets següents:

```
santi@Compute1:~$ sudo apt-get install neutron-plugin-m12 \
> neutron-plugin-openvswitch-agent
```

### **Configurar els components comuns del Networking (neutron)**

Aquests components són el mecanisme d'autenticació, l'intermediari de missatges (*message broker*) i el *plugin*. Per fer-ho, cal editar el fitxer `/etc/neutron/neutron.conf` amb el següent contingut:

- a) A la secció `[database]`, comentar qualsevol opció *connection* perquè els nodes *compute* no han d'accedir directament a la base de dades.

```
[database]
# This line MUST be changed to actually run the plugin.
# Example:
# connection = mysql://root:pass@127.0.0.1:3306/neutron
# Replace 127.0.0.1 above with the IP address of the database used by the
# main neutron server. (Leave it as is if the database runs on this host.)
# connection = sqlite:///var/lib/neutron/neutron.sqlite
# NOTE: In deployment the [database] section and its connection attribute may
# be set in the corresponding core plugin '.ini' file. However, it is suggested
# to put the [database] section and its connection attribute in this
# configuration file.
```

- b) A la secció `[DEFAULT]`, configurar el *RabbitMQ*:

```
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
rpc_backend = rabbit
```

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c) A la secció `[DEFAULT]` i `[keystone-authtoken]`, configurar l'accés al servei **Identity (keystone)**:

```
# The strategy to be used for auth.
# Supported values are 'keystone'(default), 'noauth'.
auth_strategy = keystone
```

```
[keystone_auth_token]
# auth_host = 127.0.0.1
# auth_port = 35357
# auth_protocol = http
auth_uri = http://controller:5000/v2.0
identity_uri = http://controller:35357
admin_tenant_name = service
admin_user = neutron
admin_password = 20neutron15
```

- d) A la secció [DEFAULT], habilitar el plugin *Modular Layer 2* (ML2), el servei de d'encaminament i el solapament d'IPs.

```
# (StrOpt) Neutron core plugin entrypoint to be loaded from the
# neutron.core_plugins namespace. See setup.cfg for the entrypoint names of the
# plugins included in the neutron source distribution. For compatibility with
# previous versions, the class name of a plugin can be specified instead of its
# entrypoint name.
#
core_plugin = ml2
```

```
# (ListOpt) List of service plugin entrypoints to be loaded from the
# neutron.service_plugins namespace. See setup.cfg for the entrypoint names of
# the plugins included in the neutron source distribution. For compatibility
# with previous versions, the class name of a plugin can be specified instead
# of its entrypoint name.
#
service_plugins = router
```

```
# Enable or disable overlapping IPs for subnets
# Attention: the following parameter MUST be set to False if Neutron is
# being used in conjunction with nova security groups
allow_overlapping_ips = True
```

- e) A la secció [DEFAULT], habilitar el *verbose*:

```
[DEFAULT]
# Print more verbose output (set logging level to INFO instead of default WARNING level).
verbose = True
```

### **Configurar el plugin Modular Layer 2 (ML2)**

Aquest *plugin* fa servir el mecanisme (agent) *Open vSwitch* (OVS) per facilitar una sèrie d'eines per muntar xarxes virtuals per les instàncies. Per configurar-lo, cal editar el fitxer `/etc/neutron/plugins/ml2/ml2_conf.ini` amb la següent informació:

- a) A la secció [ml2], habilitar els *drivers* de xarxa que donen suport a l'encapsulament d'encaminament tan genèric com el pla (*flat and generic routing encapsulation*):

```
[ml2]
# (ListOpt) List of network type driver endpoints to be loaded from
# the neutron.ml2.type_drivers namespace.
#
# type_drivers = local,flat,vlan,gre,vxlan
type_drivers = flat,gre
# Example: type_drivers = flat,vlan,gre,vxlan

# (ListOpt) Ordered list of network_types to allocate as tenant
# networks. The default value 'local' is useful for single-box testing
# but provides no connectivity between hosts.
#
# tenant_network_types = local
tenant_network_types = gre
# Example: tenant_network_types = vlan,gre,vxlan

# (ListOpt) Ordered list of networking mechanism driver endpoints
# to be loaded from the neutron.ml2.mechanism_drivers namespace.
mechanism_drivers = openvswitch
```

b) A la secció [ml2\_type\_gre], configurar el rang d'identificadors de túnel:

```
[ml2_type_gre]
# (ListOpt) Comma-separated list of <tun_min>:<tun_max> tuples enumerating ranges of GRE tunnel IDs
# that are available for tenant network allocation
tunnel_id_ranges = 1:1000
```

c) A la secció [securitygroup], habilitar els grups de seguretat, l'*ipset* i configurar les *iptables* de l'OVS pel *driver* del *firewall*:

```
[securitygroup]
# Controls if neutron security group is enabled or not.
# It should be false when you use nova security group.
enable_security_group = True

# Use ipset to speed-up the iptables security groups. Enabling ipset support
# requires that ipset is installed on L2 agent node.
enable_ipset = True

# OVS iptables firewall driver
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

d) A la secció [ovs], habilitar els túnels i configurar el punt d'entrada pel túnel local:

```
[ovs]
# Local instance tunnels network interface IP
local_ip = 10.0.1.31
# Tunnels enabled
enable_tunneling = True
```

e) A la secció [agent], habilitar els túnels GRE:

```
[agent]
tunnel_types = gre
```

### Configurar el servei Open vSwitch (OVS)

Només cal reiniciar el servei:

```
santi@Compute1:~$ sudo service openvswitch-switch restart
[sudo] password for santi:
openvswitch-switch stop/waiting
openvswitch-switch start/running
```

### Configurar el servei Compute (nova) perquè utilitzi el Networking (neutron)

La configuració per defecte del **nova** és utilitzar la *legacy networking*. Com que faig servir el *networking* de l'OpenStack (**neutron**), cal editar el fitxer de configuració del **nova** `/etc/nova/nova.conf`:

- A la secció [DEFAULT], configurar les APIs i els *drivers*:

```
#
# API configurations
#
network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

#
# Driver configurations
#
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSIInterfaceDriver
# Disable default nova driver, to use neutron's firewall
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

- A la secció [neutron], configurar els paràmetres d'accés

```
[neutron]
#
# Access parameters
#
url = http://controller:9696
auth_strategy = keystone
admin_auth_url = http://controller:35357/v2.0
admin_tenant_name = service
admin_username = neutron
admin_password = 20neutron15
```

### Finalització de la instal·lació

- Reiniciar el servei Compute (**nova**):

```
santi@Compute1:~$ sudo service nova-compute restart
[sudo] password for santi:
nova-compute stop/waiting
nova-compute start/running, process 4269
```

- Reiniciar l'agent de l'Open vSwitch (OVS):

```
santi@Compute1:~$ sudo service neutron-plugin-openvswitch-agent restart
neutron-plugin-openvswitch-agent stop/waiting
neutron-plugin-openvswitch-agent start/running, process 4308
```

## Verificacions

Totes les proves s'han de fer des del node Controller.

1. Carregar les credencials d'*admin*:

```
santi@Controller:~$ source admin-openrc.sh
```

2. Llistar els agents per verificar que s'estan executant correctament els agents *neutron*:

```
santi@Controller:~$ neutron agent-list
```

id	agent_type	host	alive	admin_state_up	binary
1127e434-4856-41f9-87f3-589e96e4fe3d	Open vSwitch agent	Network	:-)	True	neutron-openvswitch-agent
4321b218-b759-4fac-940b-6ecdbc4916e9	Metadata agent	Network	:-)	True	neutron-metadata-agent
9ecb5ba2-7d07-439b-a8e6-076332d362b5	DHCP agent	Network	:-)	True	neutron-dhcp-agent
c088d174-3235-46ca-8a3a-ca108c6d2d11	Open vSwitch agent	Compute1	:-)	True	neutron-openvswitch-agent
c491468f-f5a3-42b6-845c-d3867c2aa77d	L3 agent	Network	:-)	True	neutron-l3-agent



Després de realitzar les configuracions oportunes, el servei **neutron** implica els següents components:

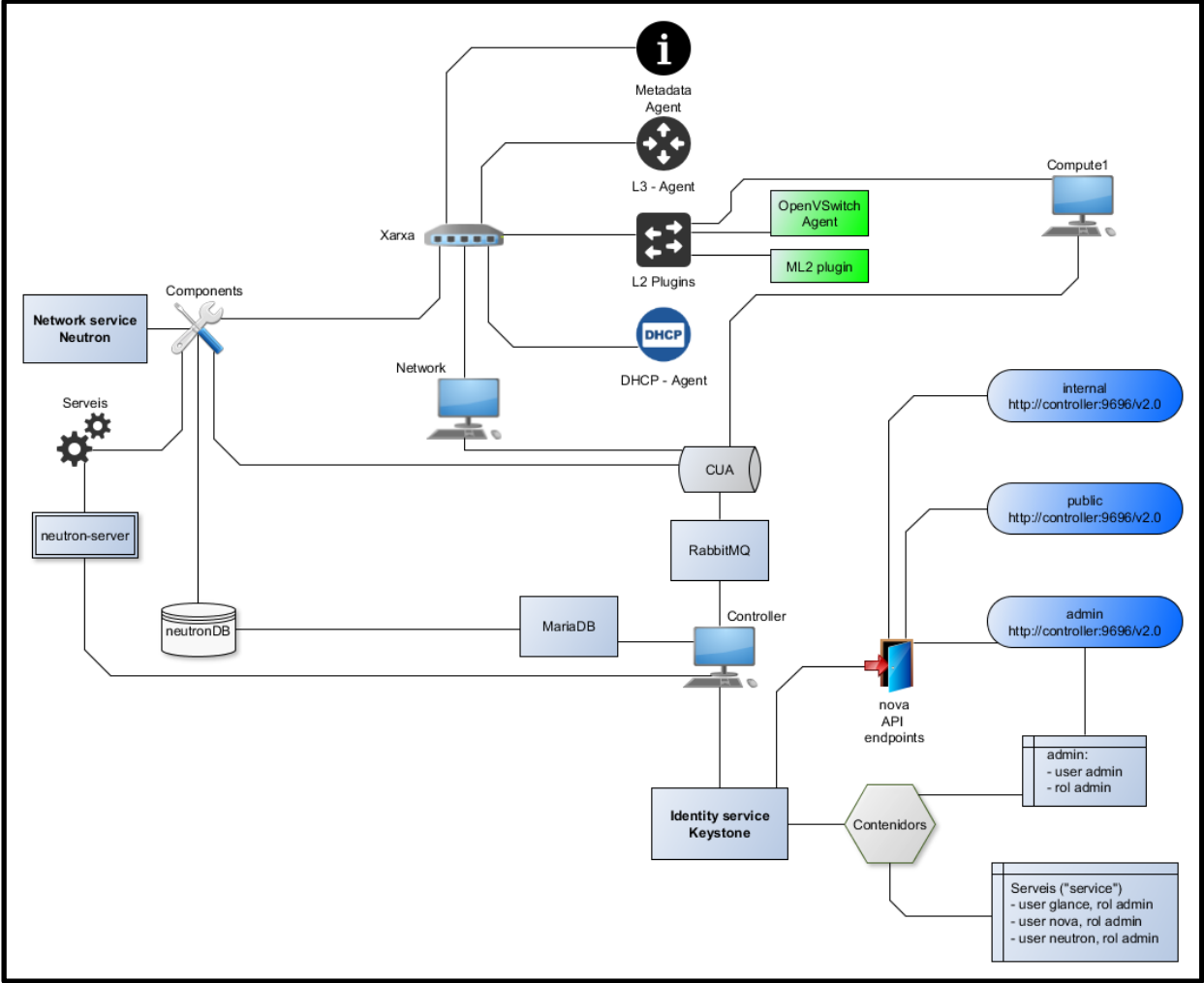


Figura 51: Components del Neutron.

## Creació de les xarxes inicials

Abans de poder executar la primera instància (VM), cal crear la infraestructura de xarxa necessària on s'haurà de connectar aquesta. Seguint la guia, la xarxa quedarà després de diverses configuracions de la següent manera:

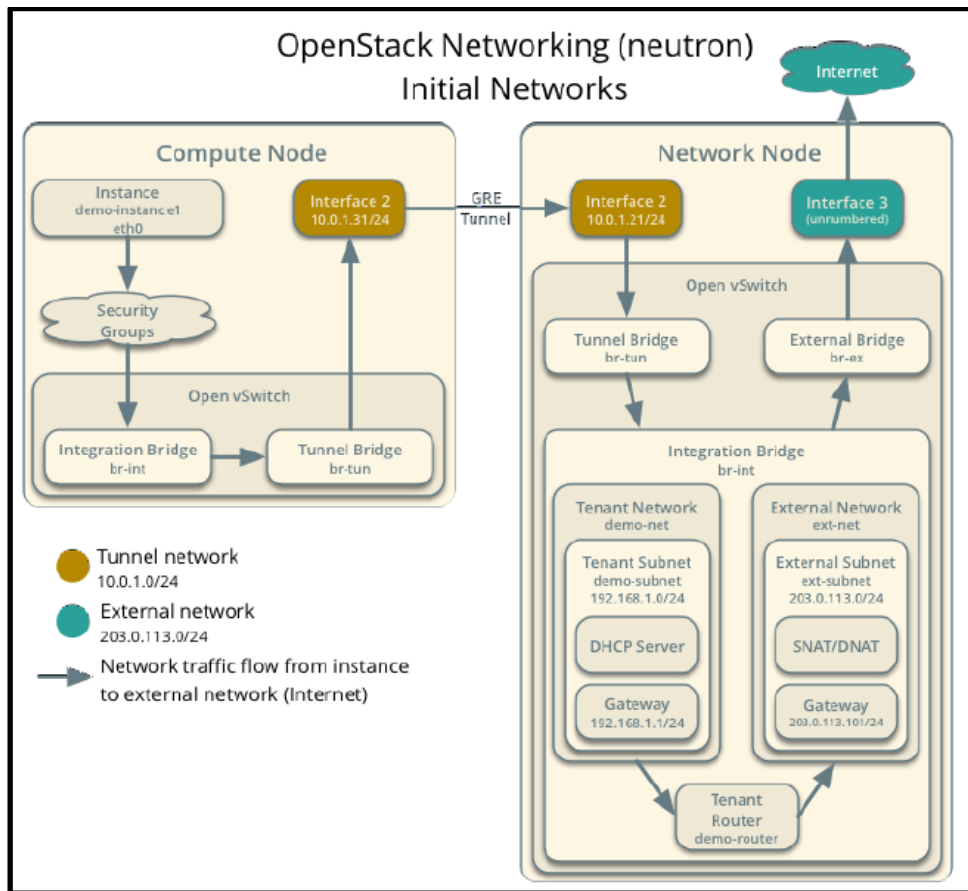


Figura 52: Xarxes inicials i bàsiques amb el Neutron

## Xarxa External

Es tracta de la xarxa que proveeix accés a Internet a les VMs. Per defecte, aquesta xarxa només admet accés a Internet des de les instàncies fent servir NAT. Es pot habilitar accés a Internet per instàncies individuals fent servir adreces IP *floating* i les *rules* necessàries al *security group* corresponent. Farem que el *tenant admin* sigui propietari d'aquesta xarxa perquè aquesta proveirà accés extern per múltiples *tenants*.

Les configuracions següents s'han de realitzar des del node **CONTROLLER**.

### Crear la xarxa External

1. Carregar les credencials d'*admin* amb l'script:

```
santi@Controller:~$ source admin-openrc.sh
```

2. Crear la xarxa:



```
santi@Controller:~$ neutron net-create ext-net --router:external True \
> --provider:physical_network external --provider:network_type flat
Created a new network:
```

Field	Value
admin_state_up	True
id	e2825034-515f-4840-991b-84de97038c15
name	ext-net
provider:network_type	flat
provider:physical_network	external
provider:segmentation_id	
router:external	True
shared	False
status	ACTIVE
subnets	
tenant_id	5a927fdf1ef24266b6c9bf555cfc5acc

### Crear una subxarxa a la xarxa External

Com si fos una xarxa física, una xarxa virtual requereix que se li assigni una subxarxa. La xarxa *External* comparteix la mateixa subxarxa i porta d'enllaç associada amb la xarxa física que hi ha connectada a la interfície del node Network. Degut a això, cal tenir en compte com s'assignen IP's a la xarxa física, ja que no s'han de produir col·lisions amb el rang d'adreces IP *floating*.

L'adreça de la xarxa *External* serà la 203.0.113.0/24 i el rang d'adreces IP *floating* anirà des de la 203.0.113.101 fins a la 203.0.113.200. La comanda que s'haurà d'executar és la següent:

```
santi@Controller:~$ neutron subnet-create ext-net --name ext-subnet \
> --allocation-pool start=203.0.113.101,end=203.0.113.200 \
> --disable-dhcp --gateway 203.0.113.1 203.0.113.0/24
Created a new subnet:
```

Field	Value
allocation_pools	{ "start": "203.0.113.101", "end": "203.0.113.200" }
cidr	203.0.113.0/24
dns_nameservers	
enable_dhcp	False
gateway_ip	203.0.113.1
host_routes	
id	797769be-6906-47c1-9323-f8fea0bf5ca1
ip_version	4
ipv6_address_mode	
ipv6_ra_mode	
name	ext-subnet
network_id	e2825034-515f-4840-991b-84de97038c15
tenant_id	5a927fdf1ef24266b6c9bf555cfc5acc

### **Xarxa tenant**

Aquesta xarxa serà la interna que connectarà les instàncies entre elles, aïllant-la d'altres xarxes que siguin d'altres *tenants*. Farem que el *tenant demo* sigui propietari d'aquesta xarxa.

### Crear la xarxa tenant

1. Crear les credencials pel contenidor demo amb l'script:

```
santi@Controller:~$ source demo-openrc.sh
```

2. Crear la xarxa:

```
santi@Controller:~$ neutron net-create demo-net
Created a new network:
+-----+
| Field          | Value                                     |
+-----+
| admin_state_up | True                                     |
| id             | d7f44344-80dd-4aff-b8b8-c482864382be   |
| name          | demo-net                                |
| router:external | False                                   |
| shared        | False                                   |
| status        | ACTIVE                                  |
| subnets      |                                          |
| tenant_id     | 55f7067fec4841b09039d7c3589855d7     |
+-----+
```

### Crear una subxarxa a la xarxa Tenant

Tal i com passava la xarxa external, aquesta xarxa també necessita una subxarxa. Es pot especificar una subxarxa vàlida (volen dir que agafi tot el rang, sense excepcions) ja que l'arquitectura aïlla les xarxes *tenant*, i per tant, aquesta subxarxa és per les màquines que hi pertanyin. La configuració de la subxarxa farà servir per defecte DHCP, així que les instàncies podran obtenir adreces IP.

Si creem la subxarxa fent servir l'@ de xarxa 192.168.1.0/24, cal executar la següent comanda:

```
santi@Controller:~$ neutron subnet-create demo-net --name demo-subnet \
> --gateway 192.168.1.1 192.168.1.0/24
Created a new subnet:
+-----+
| Field          | Value                                     |
+-----+
| allocation_pools | {"start": "192.168.1.2", "end": "192.168.1.254"} |
| cidr           | 192.168.1.0/24                           |
| dns_nameservers |                                          |
| enable_dhcp    | True                                       |
| gateway_ip     | 192.168.1.1                               |
| host_routes    |                                          |
| id            | de66dda2-2f81-4638-a64f-7afe067de198     |
| ip_version     | 4                                         |
| ipv6_address_mode |                                          |
| ipv6_ra_mode   |                                          |
| name          | demo-subnet                               |
| network_id    | d7f44344-80dd-4aff-b8b8-c482864382be   |
| tenant_id     | 55f7067fec4841b09039d7c3589855d7     |
+-----+
```

### Crear un router

El següent pas és connectar les dues xarxes per poder encaminar tràfic entre ambdues xarxes. Per fer això, caldrà un *router* virtual, el qual interconnectarà les xarxes *Tenant* i *External*. La creació d'aquest s'haurà de fer a la xarxa *Tenant*, els passos a seguir són els següents:

1. Crear el *router*.

```
santi@Controller:~$ neutron router-create demo-router
Created a new router:
+-----+-----+
| Field          | Value                               |
+-----+-----+
| admin_state_up | True                                 |
| external_gateway_info |                                     |
| id             | 5ddd914a-1ab8-4090-ac41-622e29836153 |
| name           | demo-router                         |
| routes        |                                     |
| status        | ACTIVE                              |
| tenant_id     | 55f7067fec4841b09039d7c3589855d7   |
+-----+-----+
```

2. Connectar el *router* a la subxarxa *Tenant*:

```
santi@Controller:~$ neutron router-interface-add demo-router demo-subnet
Added interface ca20cb7a-832a-424a-851b-aaca0569561c to router demo-router.
```

3. Connectar el *router* a la xarxa *External*, configurant-la com a *gateway*:

```
santi@Controller:~$ neutron router-gateway-set demo-router ext-net
Set gateway for router demo-router
```

### Verificacions de connectivitat

Abans de seguir, s'ha de verificar que es disposa de connectivitat amb la interfície del *router demo-router* que connecta amb la xarxa *External*. Per fer-ho, només cal fer ping des d'una màquina que també estigui connectada en aquesta xarxa, havent de rebre resposta. En aquest cas, dispenso d'una màquina virtual connectada en aquesta xarxa anomenada *UbuntuServer*, així la topologia és la següent:

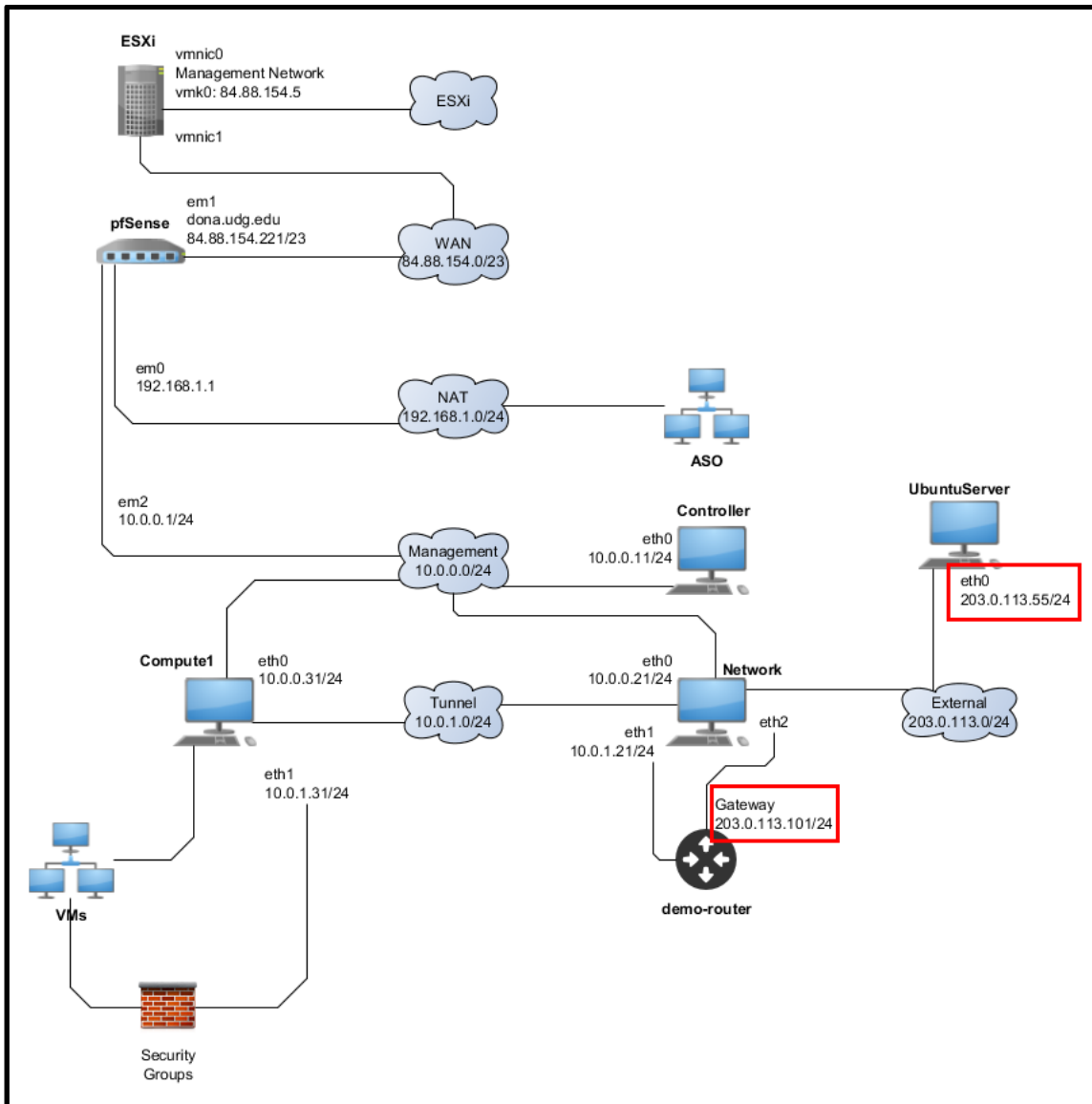


Figura 53: Escenari de proves pel Neutron

I fent la prova, aquesta té èxit:

```
santi@UbuntuServer:~$ ping -c 4 203.0.113.101
PING 203.0.113.101 (203.0.113.101) 56(84) bytes of data:
64 bytes from 203.0.113.101: icmp_seq=1 ttl=64 time=2.51 ms
64 bytes from 203.0.113.101: icmp_seq=2 ttl=64 time=0.386 ms
64 bytes from 203.0.113.101: icmp_seq=3 ttl=64 time=0.396 ms
64 bytes from 203.0.113.101: icmp_seq=4 ttl=64 time=0.308 ms
```

### 9.1.2.5 Dashboard (Horizon)

#### Requeriments del sistema

S'han de complir diversos requisits, entre ells:

- Relatiu a la instal·lació del servei **nova (Compute)**. Simplement cal tenir aquest servei per poder crear usuaris i projectes (o *tenants*): **OK**. També cal tenir a mà les URLs d'aquest servei i del **Compute**. Les del **keystone (Identity)** són les següents:
  - admin: http://controller:9292
  - internal: http://controller:9292
  - public: http://controller:9292

Les del **nova** són:

- admin: http://controller:8774/v2
  - internal: http://controller: 8774/v2
  - public: http://controller: 8774/v2
- L'usuari del servei **Identity** ha de tenir privilegis **sudo**. Com que l'Apache no serveix contingut des d'un usuari root, els usuaris han d'utilitzar el **Dashboard (horizon)** com a usuari del **keystone** amb privilegis de **root**.
- Cal tenir el Python 2.6 o el 2.7.

```
santi@Compute1:~$ readlink -f $(which python) | xargs -I % sh -c 'echo -n "%: "; % -U'
/usr/bin/python2.7: Python 2.7.6
```

- Cal instal·lar el **Dashboard** en un node on es tingui accés al **keystone**. Per tant, l'instal·laré al **Controller**.
- Informació que s'ha de donar als usuaris:
  - IP pública des d'on podran accedir al **dashboard**.
  - Usuari i contrasenya.
- Els navegadors que treballin amb el servei han de donar suport per l'HTML5 i tenir les *cookies* i el *JavaScript* habilitats.
- Si es vol accedir amb un client VNC, el navegador ha de tenir suport per HTML5 *Canvas* i *WebSockets*. Per detalls de compatibilitats pots mirar [aquí<sup>11</sup>](#) o [aquí<sup>12</sup>](#).

---

<sup>11</sup> <https://github.com/kanaka/noVNC/blob/master/README.md>

<sup>12</sup> <https://github.com/kanaka/noVNC/wiki/Browser-support>

## Instal·lació i configuració

Les següents accions s'han de realitzar des del node **Controller**.

### Instal·lació

Cal instal·lar els paquets:

```
santi@Controller:~$ sudo apt-get install openstack-dashboard \
> apache2 \
> libapache2-mod-wsgi \
> memcached \
> python-memcache
```

(\*) A l'Ubuntu s'instal·la la dependència *openstack-dashboard-ubuntu-theme*. Alguns usuaris han dit que tenen problemes amb aquest tema, si es dóna el cas, només cal eliminar aquest paquet per fer servir el tema original d'OpenStack.

### Configuració del Dashboard

Cal editar el fitxer `/etc/openstack-dashboard/local_settings.py` al node **Controller**:

- Configurar el **Dashboard (horizon)** perquè utilitzi els serveis OpenStack al node **Controller**:

```
OPENSTACK_HOST = "controller"
```

- Acceptar que qualsevol host accedeixi al **Dashboard**:

```
ALLOWED_HOSTS = ['*']
```

- Configurar el servei "*memcached session storage*" (cal comentar qualsevol altre configuració prèvia):

```
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
    }
}
```

- Configuració de la zona horària:

```
TIME_ZONE = "UTC"
```

### Finalització de la instal·lació

Reiniciar el servidor web i el servei d'emmagatzematge de sessió:

```
santi@Controller:~$ sudo service apache2 restart
* Restarting web server apache2
AH00558: apache2: Could not reliably determine the
0.11. Set the 'ServerName' directive globally to su
santi@Controller:~$ sudo service memcached restart
Restarting memcached: memcached.
```

## Verificacions

- Accedir al **Dashboard** a través d'un navegador: <http://controller/horizon>.

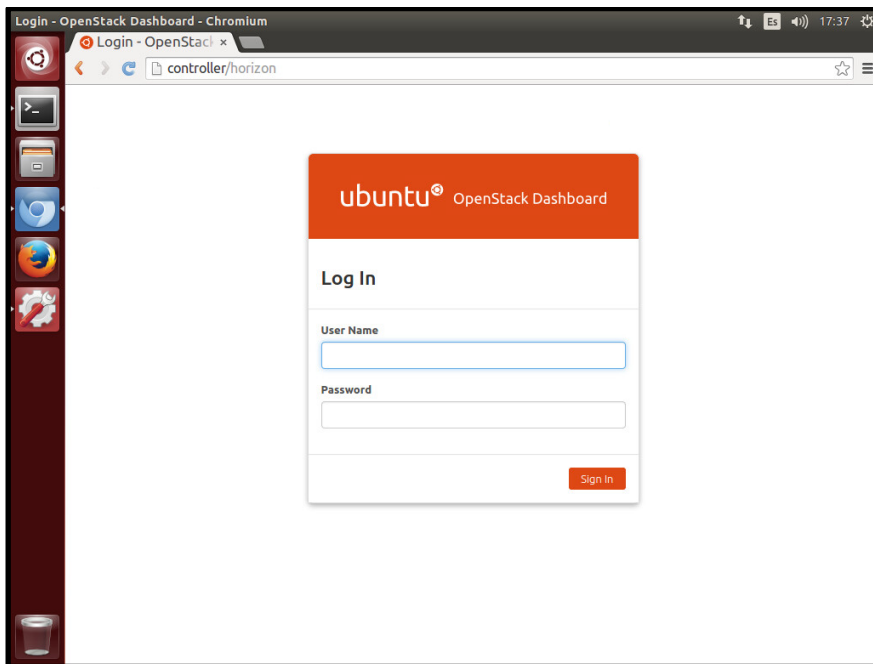


Figura 54: Login del Dashboard

- Autenticar-se mitjançant les credencials d'**admin** o bé les del **demo** (usuaris del **keystone**).

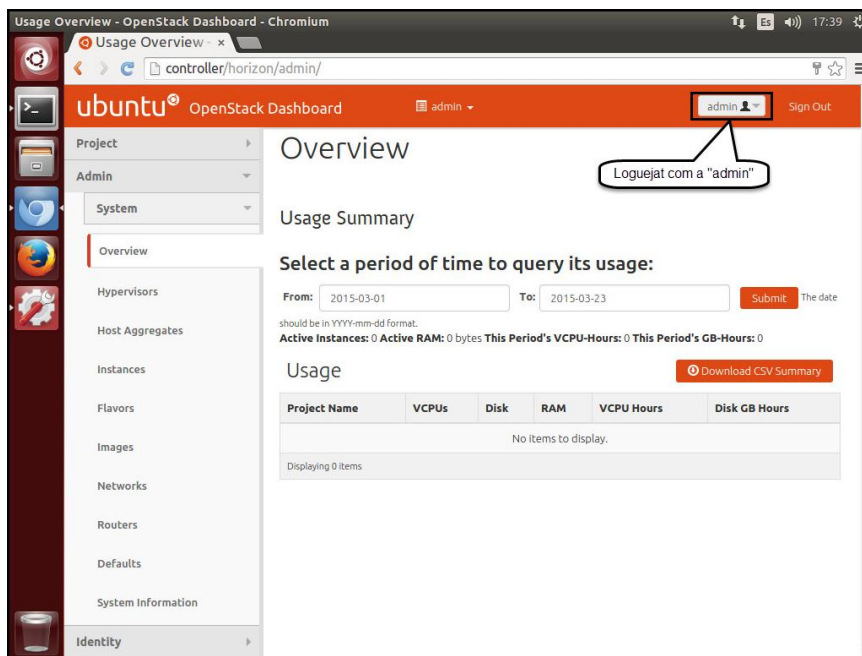


Figura 55: Vista d'un administrador

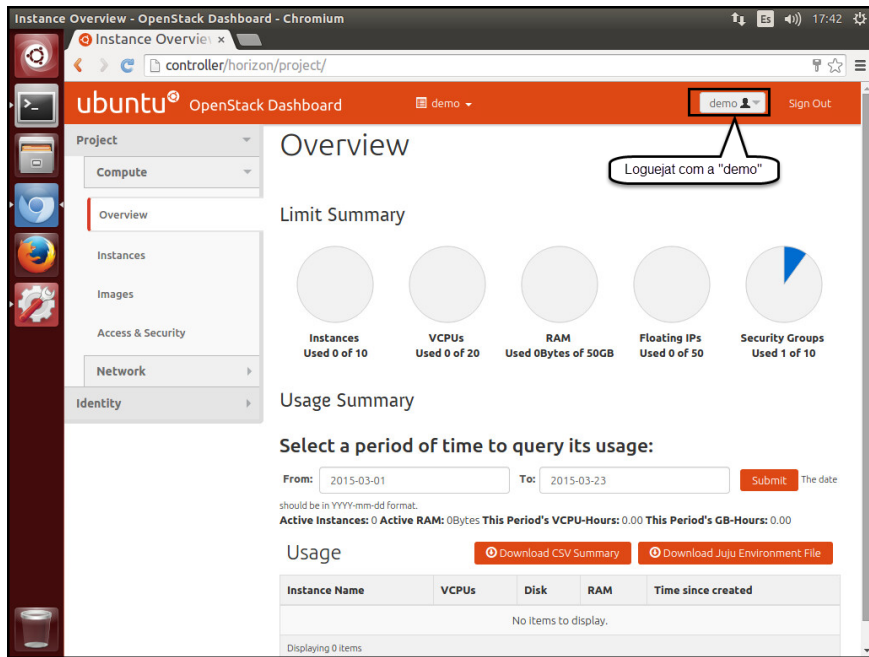


Figura 56: Vista d'un usuari

Aquests accessos estan fets des d'una màquina UbuntuDesktop muntada dins la pròpia xarxa Manage:

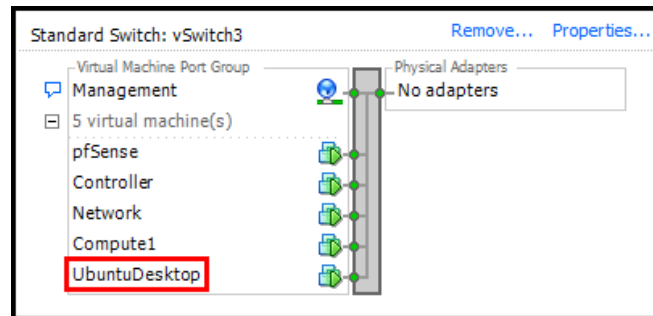


Figura 57: Ubicació de la màquina UbuntuDesktop



### 9.1.2.6 Orchestration module (Heat)

#### Instal·lació i configuració

Tot el procés que es descriurà a continuació, s'haurà de realitzar al node **Controller**.

#### Prerequisites

1. Creació de la base de dades.
  - a. Accedir a l'administració del **mysql**:

```
santi@Controller:/home$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 9639
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

- b. Crear la base de dades **heat**:

```
MariaDB [(none)]> CREATE DATABASE heat;
Query OK, 1 row affected (0.00 sec)
```

- c. Garantir els privilegis d'accés a la base de dades:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'localhost' \
-> IDENTIFIED BY '20heat15';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'%' \
-> IDENTIFIED BY '20heat15';
Query OK, 0 rows affected (0.00 sec)
```

2. Carregar les credencials d'admin.

```
santi@Controller:~$ source admin-openrc.sh
```

3. Crear les credencials pel servei.

- a. Crear l'usuari **heat**:

```
santi@Controller:~$ keystone user-create \
> --name heat \
> --pass 20heat15;
+-----+-----+
| Property | Value |
+-----+-----+
| email    |      |
| enabled  | True  |
| id       | 33fb89b1e9cc47bf3ba75aedc01c1d4 |
| name     | heat  |
| username | heat  |
+-----+-----+
```

- b. Afegir el rol **admin** a l'usuari **heat**:

```
santi@Controller:~$ keystone user-role-add \  
> --user heat \  
> --tenant service \  
> --role admin;
```

- c. Crear el rol **heat\_stack\_owner**:

```
santi@Controller:~$ keystone role-create \  
> --name heat_stack_owner;  
  
+-----+  
| Property | Value |  
+-----+  
| id       | 3ddc5eac73c34191af76980fbb9973ea |  
| name     | heat_stack_owner |  
+-----+
```

- d. Afegir el rol **heat\_stack\_owner** al contenidor **demo** i també a l'usuari **demo** (s'ha d'afegir aquest rol per cada usuari que hagi de poder administrar conjunts de recursos):

```
santi@Controller:~$ keystone user-role-add \  
> --user demo \  
> --tenant demo \  
> --role heat_stack_owner;
```

- e. Crear el rol **heat\_stack\_user** (Per defecte el **Heat** assigna aquest rol als usuaris que aquest crea durant la creació d'un stack de recursos. Aquest rol aplica restriccions sobre l'API, així que cal evitar de tenir-lo per usuaris que hagin de tenir el rol d'**owner**.):

```
santi@Controller:~$ keystone role-create \  
> --name heat_stack_user;  
  
+-----+  
| Property | Value |  
+-----+  
| id       | 275d58d7b06f4fc9b9dfd8e070421de2 |  
| name     | heat_stack_user |  
+-----+
```

- f. Crear les entitats **heat** i **heat-cfn**:

```
> --description "Orchestration";  
  
+-----+  
| Property | Value |  
+-----+  
| description | Orchestration |  
| enabled     | True |  
| id         | ced639e5ab2e47e6a7130ed3af0f6f08 |  
| name       | heat |  
| type       | orchestration |  
+-----+
```

```
santi@Controller:~$ keystone service-create \
> --name heat-cfn \
> --type cloudformation \
> --description "Orchestration";
```

Property	Value
description	Orchestration
enabled	True
id	bafb52ecf12c4a5db5d9f8cd4fedad5a
name	heat-cfn
type	cloudformation

4. Crear els punts d'entrada a l'API del servei.

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ orchestration / {print $2}') \
> --publicurl http://controller:8004/v1/\^(tenant_id)s \
> --internalurl http://controller:8004/v1/\^(tenant_id)s \
> --adminurl http://controller:8004/v1/\^(tenant_id)s \
> --region regionOne;
```

Property	Value
adminurl	http://controller:8004/v1/\^(tenant_id)s
id	b68f50782a3b444e956e9d3058874417
internalurl	http://controller:8004/v1/\^(tenant_id)s
publicurl	http://controller:8004/v1/\^(tenant_id)s
region	regionOne
service_id	ced639e5ab2e47e6a7130ed3af0f6f08

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ cloudformation / {print $2}') \
> --publicurl http://controller:8000/v1 \
> --internalurl http://controller:8000/v1 \
> --adminurl http://controller:8000/v1 \
> --region regionOne;
```

Property	Value
adminurl	http://controller:8000/v1
id	07f6da3ac8b94d7a815c4c4a845b632e
internalurl	http://controller:8000/v1
publicurl	http://controller:8000/v1
region	regionOne
service_id	bafb52ecf12c4a5db5d9f8cd4fedad5a

## Instal·lació dels components del Heat

1. Instal·lació dels paquets:

```
santi@Controller:~$ sudo apt-get install \
> heat-api \
> heat-api-cfn \
> heat-engine \
> python-heatclient;
```

2. Editar el fitxer de configuració (/etc/heat/heat.conf):

a. A la secció [database], configurar l'accés a la **base de dades**:

```
# The SQLAlchemy connection string to use to connect to the
# database. (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql://heat:20heat15@controller/heat
```

- b. A la secció [DEFAULT], configurar l'accés al servei RabbitMQ:

```
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
rpc_backend=rabbit
```

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c. A les seccions [keystone\_authtoken] i [ec2authtoken], configurar l'accés al keystone:

```
[keystone_authtoken]
# Complete public Identity API endpoint (string value)
auth_uri = http://controller:5000/v2.0
# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri = http://controller:35357
# Keystone service account tenant name to validate user tokens
# (string value)
admin_tenant_name = service
# Keystone account username (string value)
admin_user = heat
# Keystone account password (string value)
admin_password = 20heat15

[ec2authtoken]

#
# Options defined in heat.api.aws.ec2token
#

# Authentication Endpoint URI. (string value)
auth_uri = http://controller:5000/v2.0
```

- d. A la secció [DEFAULT], configurar les URLs de metadades i *waitcondition*:

```
# URL of the Heat metadata server. (string value)
heat_metadata_server_url = http://controller:8000

# URL of the Heat waitcondition server. (string value)
heat_waitcondition_server_url = http://controller:8000/v1/waitcondition
```

- e. Habilitar el *verbose* a la secció [DEFAULT], també:

```
# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose=false
```

3. Escriure els canvis a la base de dades del **heat**:

```
santi@Controller:~$ sudo su -s /bin/sh -c "heat-manage db sync" heat
```

## Finalització de la instal·lació

1. Reiniciar els serveis de l'Orchestration:

```
santi@Controller:~$ sudo service heat-api restart
heat-api stop/waiting
heat-api start/running, process 5210
santi@Controller:~$ sudo service heat-api-cfn restart
heat-api-cfn stop/waiting
heat-api-cfn start/running, process 5227
santi@Controller:~$ sudo service heat-engine restart
stop: Unknown instance:
heat-engine start/running, process 5247
```

2. Aquest pas no està a la guia. Com es pot veure en el pas anterior, el *heat-engine* no estava engegat. Una de les coses que m'he trobat és que a diferència de la instal·lació de la resta de serveis de l'OpenStack, aquest servei no s'arrenca automàticament a l'inici. Per tant, s'ha de fer *hand-made*:

```
santi@Controller:~$ sudo update-rc.d heat-engine defaults
```

Més endavant, quan estigui muntant el servei **sahara**, em trobaré amb que aquesta no és la solució ideal. El motiu és que fent això, el que s'aconsegueix és definir el *heat-engine* com a servei del sistema operatiu, quan en realitat això no és així. Aquest procés forma part del software de l'usuari, i per tant, ha de ser tractat com a tal. En conseqüència, s'ha de fer servir [l'Upstart<sup>13</sup>](#) per tractar amb ell, cosa que ja està feta amb la instal·lació del paquet, però s'han deixat un petit detall: afegir la comanda *respawn* al fitxer d'inici */etc/init/heat-engine.conf*. Aquesta comanda, el que fa és reiniciar el servei automàticament en cas que aquest s'hagi aturat. Així, la feina descrita es pot observar al requadre vermell:

---

<sup>13</sup> <https://help.ubuntu.com/community/UbuntuBootupHowto>

```
description "Heat Engine server"
author "James Page <james.page@ubuntu.com>"

start on runlevel [2345]
stop on runlevel [!2345]

respawn

exec start-stop-daemon --start --chuid heat --exec /usr/bin/heat-engine
```

I si comparem amb el fitxer d'inici del **keystone** a `/etc/init/keystone.conf`:

```
description "Keystone API server"
author "Soren Hansen <soren@linux2go.dk>"

start on runlevel [2345]
stop on runlevel [!2345]

respawn

exec start-stop-daemon --start --chuid keystone \
    --chdir /var/lib/keystone --name keystone \
    --exec /usr/bin/keystone-all
```

3. Eliminar la base de dades SQLite, ja que es fa servir el MySQL:

```
santi@Controller:~$ rm -f /var/lib/heat/heat.sqlite
```

## Verificacions

1. Carregar les credencials del *tenant* demo:

```
santi@Controller:~$ source demo-openrc.sh
```

2. El **heat** fa servir *templates* per descriure els *stacks*. Per més informació sobre el seu llenguatge, ves [aquí](http://docs.openstack.org/developer/heat/template_guide/index.html)<sup>14</sup>. Si vols [més documentació](http://docs.openstack.org/developer/heat/template_guide/index.html)<sup>15</sup>. Per provar, farem un *template* de test anomenat *test-stack.yml*:

---

<sup>14</sup> [http://docs.openstack.org/developer/heat/template\\_guide/index.html](http://docs.openstack.org/developer/heat/template_guide/index.html)

<sup>15</sup> [http://docs.openstack.org/developer/heat/template\\_guide/index.html](http://docs.openstack.org/developer/heat/template_guide/index.html)

```

heat_template_version: 2014-10-16
description: A simple server.

parameters:
  ImageID:
    type: string
    description: Image use to boot a server
  NetID:
    type: string
    description: Network ID for the server

resources:
  server:
    type: OS::Nova::Server
    properties:
      image: { get_param: ImageID }
      flavor: m1.tiny
      networks:
        - network: { get_param: NetID }

outputs:
  private_ip:
    description: IP address of the server in the private network
    value: { get_attr: [ server, first_address ] }

```

3. Crear un *stack* amb el *template* que acabo de crear:

```

santi@Controller:~$ heat stack-create \
> -f test-stack.yml \
> -P "ImageID=cirros-0.3.3-x86_64;NetID=$NET_ID" testStack

```

id	stack_name	stack_status	creation_time
257ed823-f5c8-45fa-ab97-efcd46a9dd95	testStack	CREATE_IN_PROGRESS	2015-04-14T16:57:14Z

4. Verificar que s'ha creat l'*stack*:

```

santi@Controller:~$ heat stack-list

```

id	stack_name	stack_status	creation_time
257ed823-f5c8-45fa-ab97-efcd46a9dd95	testStack	CREATE_FAILED	2015-04-14T16:57:14Z

Després d'haver fet la instal·lació del **Heat** seguint la guia de l'OpenStack pel Juno, vaig fer les verificacions. Com es pot veure en l'apartat dedicat a aquestes (l'anterior), en el pas 4, quan llisto els diferents *stacks* (conjunt de recursos del cloud) es pot observar que no ha funcionat:

```

santi@Controller:~$ heat stack-list

```

id	stack_name	stack_status	creation_time
257ed823-f5c8-45fa-ab97-efcd46a9dd95	testStack	CREATE_FAILED	2015-04-14T16:57:14Z

Malgrat haver fet aquesta verificació, en el moment no em vaig adonar i ho vaig ignorar. Així que després d'uns dies de parada, quan vaig tornar a reenganxar amb el projecte, la primera prova que vaig voler fer va ser crear un *stack* des de l'aplicació

web, trobant-me que no funcionava. Després d'intentar-ho diversos cops, no ho vaig aconseguir i vaig tornar a revisar la documentació que havia escrit, trobant-me amb el problema que acabo d'explicar.

Vaig començar a revisar i llegir tota la documentació oficial, però no vaig trobar res que em guiés. Així, vaig optar per mirar al fòrum oficial per veure si algú havia tingut algun problema semblant, i com sempre, hi ha hagut algú que li ha passat i [ha preguntat](#)<sup>16</sup>. Doncs bé, les proves realitzades van ser:

- Intentar tornar a crear l'*stack* des de la CLI (*Command Line Interface*) del node **Controller**. **No va funcionar**.
- Fer una màquina virtual des de la CLI i arrencar-la. **No va funcionar**.
- Fer les dues accions anteriors des de l'aplicació web. **No va funcionar**.
- Intentar arrencar la màquina virtual que vaig crear inicialment. **Funcionava**.

Arribat aquest punt, veient que la primera VM funcionava, havia de provar si realment funcionava, accedint amb l'hypervisor des de l'aplicació web, i efectivament, aquesta funcionava.

Finalment, vaig llistar els serveis relacionats amb el **Nova** (*source admin.sh, nova service list*), per veure què estava passant, i el **nova-compute** no estava arrencat, l'**scheduler** tampoc... Vaig donar amb el problema. La causa d'aquest ha sigut el reinici accidental en alguna ocasió del node **Controller** (apretar **ctrl+alt+ins** a la consola de l'vSphere). Així, la solució del problema ha estat parar els nodes i arrencar-los en aquest ordre:

1. Controller.
2. Network.
3. Compute1.

Un cop arrencats d'aquesta manera i passat un temps prudencial, he començat a provar la creació i execució d'VMs, funcionant perfectament.

Abans de tornar a provar la creació d'*stacks* amb fitxers, he volgut comprovar la connectivitat de les màquines virtuals. La connectivitat des de la xarxa External no és cap problema, ja que gràcies al servei **neutron**, és tan senzill com assignar una *floating IP* a la VM des de l'aplicació web i ja està. El problema ha sigut poder connectar la màquina cap a fora.

Per poder provar la connectivitat de les VM cap a fora, el que he fet en primera instància ha sigut provar amb una VM de proves a l'ESXi fer un *router*. Aquesta VM disposava com a SO d'un Ubuntu Server, pel qual he realitzat les configuracions adients per treballar com a *router* dins l'escenari que tinc muntat. **No va funcionar** de

---

<sup>16</sup> <https://ask.openstack.org/en/question/53250/heat-stack-create-is-failing-with-no-valid-host-error/>



cap de les maneres; vaig provar amb les VMs de l'OpenStack i no funcionava, vaig provar amb VMs de prova a l'ESXi i tampoc. Es tractava del mateix problema redundant que em vaig trobar quan volia connectivitat des de la xarxa Management cap a fora. Finalment la solució per donar connectivitat des de la xarxa External cap a fora, ha estat com llavors, afegir una nova interfície al pfSense i realitzar les mateixes configuracions que es van fer, quedant l'escenari d'aquesta manera:

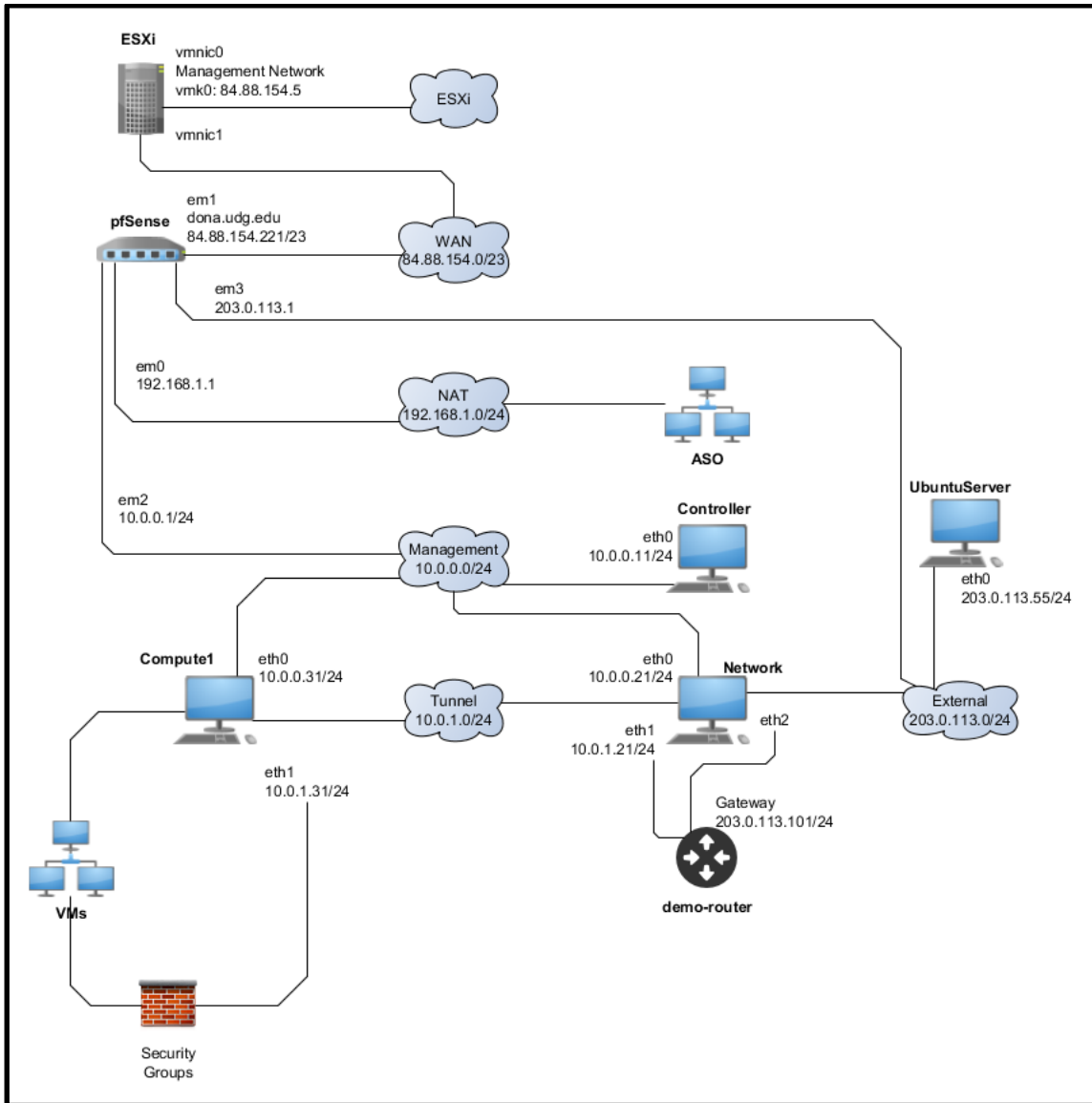


Figura 58: Escenari amb la nova connectivitat de la xarxa External al pfSense

Tornant a l'origen, he estat fent totes aquestes comprovacions perquè la creació de VMs a través de *templates* no funcionava (Orchestration - Heat). Així que torno a provar crear *stacks* amb el fitxer de *template* i per fi funciona:

```
santi@Controller:~$ heat stack-create \
> -f test-stack.yml \
> -P "ImageID=$IMAGE;NetID=$NET_ID" \
> testStack
+-----+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+-----+
| 68371ede-1a77-4c23-8b76-ff637c0894a0 | testStack | CREATE_IN_PROGRESS | 2015-04-21T08:22:43Z |
+-----+-----+-----+-----+
santi@Controller:~$ heat stack-list
+-----+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+-----+
| 68371ede-1a77-4c23-8b76-ff637c0894a0 | testStack | CREATE_COMPLETE | 2015-04-21T08:22:43Z |
+-----+-----+-----+-----+
```

I si ho miro a través del navegador amb l'Horizon:

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created
<input type="checkbox"/>	testStack-server-6savs47bxjjq	cirros-0.3.3-x86_64	192.168.1.7	m1.tiny	-	Active	nova	None	Running	4 minutes
<input type="checkbox"/>	demo-instance1	cirros-0.3.3-x86_64	192.168.1.6 203.0.113.102	m1.tiny	demo-key	Active	nova	None	Running	5 days, 19 hours

### 9.1.2.7 Telemetry module (Ceilometer)

#### Instal·lació i configuració al node Controller

##### Prerequisits

Cal instal·lar els paquets de *MongoDB*, crear una base de dades *MongoDB*, configurar les credencials del servei amb el **keystone** i configurar els punts d'entrada a la seva API (API endpoints).

1. Instal·lar els paquets de la DB MongoDB:

```
santi@Controller:~$ sudo apt-get install \
> mongodb-server \
> mongodb-clients \
> python-pymongo
```

2. Editar el fitxer de configuració */etc/mongodb.conf*:

- a. Posar la IP del node **Controller** a la xarxa Management (10.0.0.11) al paràmetre *bind\_ip*:

```
bind_ip = 10.0.0.11
```

- b. Per defecte, el *MongoDB* utilitza diversos fitxers d'1GB per fer *journaling*, al directori */var/lib/mongodb/journal*. Per reduir la mida de cada fitxer a 128MB i limitar el consum total d'espai a 512MB, cal posar el paràmetre *smallfiles* a *true*:

```
#Journaling files to 128mb
smallfiles = true
```

- c. Cal reiniciar el servei, però a més cal eliminar els fitxers de *journaling* previs:

```
santi@Controller:~$ sudo service mongod stop
mongod stop/waiting
santi@Controller:~$ sudo rm /var/lib/mongod/journal/prealloc.*
santi@Controller:~$ sudo service mongod start
mongod start/running, process 4357
```

3. Crear la base de dades *ceilometer*:

```
santi@Controller:~$ sudo mongo --host controller --eval '
> db = db.getSiblingDB("ceilometer");
> db.addUser({user: "ceilometer",
> pwd: "20ceilometer15",
> roles: [ "readWrite", "dbAdmin"]})'
MongoDB shell version: 2.4.9
connecting to: controller:27017/test
{
  "user" : "ceilometer",
  "pwd" : "84ed7a26489267636d2384a222620bc0",
  "roles" : [
    "readWrite",
    "dbAdmin"
  ],
  "_id" : ObjectId("5537c33e8f6970cb0afe94c7")
}
```

4. Carregar les credencials d'admin:

```
santi@Controller:~$ source admin-openrc.sh
```

5. Crear les credencials pel servei al **keystone**:

- a. Crear l'usuari *ceilometer*:

```
santi@Controller:~$ keystone user-create \
> --name ceilometer \
> --pass 20ceilometer15
+-----+
| Property | Value |
+-----+
| email    |       |
| enabled  | True  |
| id       | 1c793ac28b644d7b9607bf3a35161082 |
| name     | ceilometer |
| username | ceilometer |
+-----+
```

- b. Afegir el rol *admin* a l'usuari *ceilometer*:

```
santi@Controller:~$ keystone user-role-add \
> --user ceilometer \
> --tenant service \
> --role admin
```

- c. Crear l'entitat pel servei *ceilometer*:

```
santi@Controller:~$ keystone service-create \
> --name ceilometer \
> --type metering \
> --description "Telemetry"
```

Property	Value
description	Telemetry
enabled	True
id	bb935fa98f654c03aec02acde5930587
name	ceilometer
type	metering

6. Crear els punts d'entrada a l'API:

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ metering / {print $2}') \
> --publicurl http://controller:8777 \
> --internalurl http://controller:8777 \
> --adminurl http://controller:8777 \
> --region regionOne
```

Property	Value
adminurl	http://controller:8777
id	934ff05aed004d4e843d429ff59b20e9
internalurl	http://controller:8777
publicurl	http://controller:8777
region	regionOne
service_id	bb935fa98f654c03aec02acde5930587

### **Instal·lar i configurar els paquets del mòdul Telemetry**

1. Instal·lar els paquets:

```
santi@Controller:~$ sudo apt-get install \
> ceilometer-api \
> ceilometer-collector \
> ceilometer-agent-central \
> ceilometer-agent-notification \
> ceilometer-alarm-evaluator \
> ceilometer-alarm-notifier \
> python-ceilometerclient_
```

2. Generar un valor aleatori per fer-lo servei com a secret pel *metering*:

```
santi@Controller:~$ sudo openssl rand -hex 10
c7f1188eaec8cbd069ba
```

3. Editar el fitxer de configuració */etc/ceilometer/ceilometer.conf*:

- a. A la secció [database], configurar-ne l'accés:

```
# The SQLAlchemy connection string to use to connect to the
# database. (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mongodb://ceilometer:20ceilometer15@controller:27017/ceilometer
```

- b. A la secció [DEFAULT], configurar l'accés pel *RabbitMQ*:

```
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
rpc_backend=rabbit
```

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c. A les seccions [DEFAULT] i [keystone\_authtoken], configurar l'accés al servei **keystone** (cal comentar qualsevol línia *auth\_host*, *auth\_port* i *auth\_protocol*, ja que l'*identity\_uri* les substitueix):

```
# Authenticate way
auth_strategy = keystone
```

```
# Complete public Identity API endpoint (string value)
auth_uri = http://controller:5000/v2.0

# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri = http://controller:35357
```

```
# Keystone account username (string value)
admin_user = ceilometer

# Keystone account password (string value)
admin_password = 20ceilometer15

# Keystone service account tenant name to validate user tokens
# (string value)
admin_tenant_name = service
```

- d. A la secció [servicecredentials], configurar les credencials pel servei:

```

# User name to use for OpenStack service access. (string
# value)
os_username = ceilometer

# Password to use for OpenStack service access. (string value)
os_password = 20ceilometer15

# Tenant ID to use for OpenStack service access. (string
# value)
#os_tenant_id=

# Tenant name to use for OpenStack service access. (string
# value)
os_tenant_name = service

# Certificate chain for SSL validation. (string value)
#os_cacert=<None>

# Auth URL to use for OpenStack service access. (string value)
os_auth_url = http://controller:5000/v2.0

```

- e. A la secció [publisher], configurar el secret pel *metering* (obtingut prèviament):

```

# Secret value for signing metering messages. (string value)
# Deprecated group/name - [DEFAULT]/metering_secret
# Deprecated group/name - [publisher_rpc]/metering_secret
metering_secret = c7f1188eaec8cbd069ba

```

- f. Habilitar el *verbose* a la secció [DEFAULT]:

```

# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose = false

```

### **Finalització de la instal·lació**

Reiniciar els serveis del *Telemetry*:

```
santi@Controller:~$ sudo service ceilometer-agent-central restart
[sudo] password for santi:
ceilometer-agent-central stop/waiting
ceilometer-agent-central start/running, process 6547
santi@Controller:~$ sudo service ceilometer-agent-notification restart
ceilometer-agent-notification stop/waiting
ceilometer-agent-notification start/running, process 6575
santi@Controller:~$ sudo service ceilometer-api restart
ceilometer-api stop/waiting
ceilometer-api start/running, process 6598
santi@Controller:~$ sudo service ceilometer-collector restart
ceilometer-collector stop/waiting
ceilometer-collector start/running, process 6620
santi@Controller:~$ sudo service ceilometer-alarm-evaluator restart
ceilometer-alarm-evaluator stop/waiting
ceilometer-alarm-evaluator start/running, process 6643
santi@Controller:~$ sudo service ceilometer-alarm-notifier restart
ceilometer-alarm-notifier stop/waiting
ceilometer-alarm-notifier start/running, process 6665
```

## Instal·lació i configuració dels nodes de Computació

El mòdul fa servir una combinació de notificacions i un agent per recollir les mètriques de Computació. Les següents configuracions caldrà fer-les a cadascun dels nodes de computació, en aquest cas només serà al node **Compute1**.

### Instal·lar i configurar l'agent

1. Instal·lar els paquets necessaris:

```
santi@Compute1:~$ sudo apt-get install ceilometer-agent-compute
```

2. Editar el fitxer de configuració `/etc/ceilometer/ceilometer.conf`:
  - a. A la secció `[publisher]`, configurar els secret de *metering* que he generat anteriorment:

```
# Secret value for signing metering messages. (string value)
# Deprecated group/name - [DEFAULT]/metering_secret
# Deprecated group/name - [publisher_rpc]/metering_secret
metering_secret = c7f1188eaec8cbd069ba
```

- b. A la secció `[DEFAULT]`, configurar l'accés al *RabbitMQ*:

```
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host = controller
```

```
# The RabbitMQ password. (string value)
rabbit_password = 20openstack15
```

- c. A la secció `[keystone_auth token]`, configurar l'accés al **keystone**:



```
# Complete public Identity API endpoint (string value)
auth_uri = http://controller:5000/v2.0

# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri = http://controller:35357
```

```
# Keystone account username (string value)
admin_user = ceilometer

# Keystone account password (string value)
admin_password = 20ceilometer15

# Keystone service account tenant name to validate user tokens
# (string value)
admin_tenant_name = service
```

- d. A la secció [service\_credentials], configurar les credencials del servei:

```
# Auth URL to use for OpenStack service access. (string value)
os_auth_url = http://controller:5000/v2.0
```

```
# User name to use for OpenStack service access. (string
# value)
os_username = ceilometer

# Password to use for OpenStack service access. (string value)
os_password = 20ceilometer15
```

```
# Auth URL to use for OpenStack service access. (string value)
os_auth_url = http://controller:5000/v2.0
```

```
# Region name to use for OpenStack service endpoints. (string
# value)
os_region_name = regionOne

# Type of endpoint in Identity service catalog to use for
# communication with OpenStack services. (string value)
os_endpoint_type = internalURL
```

- e. A la secció [DEFAULT], habilitar el *verbose*:

```
# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose = true
```

### **Configurar les notificacions**

Configurar el **nova** perquè envii notificacions al bus de missatges. Per fer això caldrà Editar el fitxer de configuració del **nova** */etc/nova/nova.conf* amb el següent contingut a la secció [DEFAULT]:



```
#  
# Telemetry (ceilometer)  
#  
instance_usage_audit = True  
instance_usage_audit_period = hour  
notify_on_state_change = vm_and_task_state  
notification_driver = messagingv2
```

### Finalització de la instal·lació

1. Reiniciar l'agent:

```
santi@Compute1:~$ sudo service ceilometer-agent-compute restart  
ceilometer-agent-compute stop/waiting  
ceilometer-agent-compute start/running, process 30457
```

2. Reiniciar el servei nova:

```
santi@Compute1:~$ sudo service nova-compute restart  
nova-compute stop/waiting  
nova-compute start/running, process 30542
```

### Configurar el servei d'Image (glance)

Per obtenir esdeveniments i mostres sobre les imatges, cal configurar el **glance** perquè envii notificacions a través del bus de missatges. Cal realitzar les següents configuracions al node **Controller**.

1. Editar el fitxer de configuració de l'api i del registre del **glance**, ubicats a `/etc/glance/glance-api.conf` i a `/etc/glance/glance-registry.conf` respectivament, afegint a la secció [DEFAULT] d'ambdós el següent contingut:

```
# Driver or drivers to handle sending notifications. Set to  
# 'messaging' to send notifications to a message queue.  
notification_driver = messagingv2
```

```
# Messaging driver used for 'messaging' notifications driver  
rpc_backend = rabbit
```

```
rabbit_host = controller
```

```
rabbit_password = 20openstack15
```

2. Reiniciar el servei d'Image (glance):

```
santi@Controller:~$ sudo service glance-registry restart  
glance-registry stop/waiting  
glance-registry start/running, process 23710  
santi@Controller:~$ sudo service glance-api restart  
glance-api stop/waiting  
glance-api start/running, process 23728
```

## Verificacions

S'han de realitzar des del **Controller**.

1. Carregar les credencials de l'admin:

```
santi@Controller:~$ source admin-openrc.sh
```

2. Llistar les mètriques disponibles:

```
santi@Controller:~$ ceilometer meter-list
+-----+-----+-----+-----+-----+-----+
| Name      | Type | Unit | Resource ID                               | User ID | Project ID |
+-----+-----+-----+-----+-----+-----+
| image     | gauge | image | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | None    | 5a927fdf1ef24266b6c9bf555cfc5acc |
| image.size | gauge | B     | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | None    | 5a927fdf1ef24266b6c9bf555cfc5acc |
+-----+-----+-----+-----+-----+-----+
```

3. Descarregar una imatge del servei Image:

```
santi@Controller:~$ glance image-download "cirros-0.3.3-x86_64" > cirros.img
```

4. Llistar les mètriques disponibles una altra vegada per validar la detecció de la descàrrega que s'ha fet de la imatge:

```
santi@Controller:~$ ceilometer meter-list
+-----+-----+-----+-----+-----+-----+
| Name      | Type | Unit | Resource ID                               | User ID | Project ID |
+-----+-----+-----+-----+-----+-----+
| image     | gauge | image | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | None    | 5a927fdf1ef24266b6c9bf555cfc5acc |
| image.download | delta | B     | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | ee3eb8869a1a44738e166b23f3241cf6 | 5a927fdf1ef24266b6c9bf555cfc5acc |
| image.serve | delta | B     | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | None    | 5a927fdf1ef24266b6c9bf555cfc5acc |
| image.size | gauge | B     | 7585c3a8-cbdb-4c9b-b819-f819331f95e9 | None    | 5a927fdf1ef24266b6c9bf555cfc5acc |
+-----+-----+-----+-----+-----+-----+
```

5. Recuperar les estadístiques d'ús des de la mètrica *image.download*:

```
santi@Controller:~$ ceilometer statistics -m image.download
+-----+-----+-----+-----+-----+-----+
| Period | Period Start | Period End | Max | Min | Avg |
| Sum   | Count | Duration | Duration Start | Duration End | Duration |
+-----+-----+-----+-----+-----+-----+
| 0     | 2015-04-23T15:26:13.000768 | 2015-04-23T15:26:13.000768 | 13200896.0 | 13200896.0 | 13200896.0 |
| 1     | 0.0 | 2015-04-23T15:26:13.000768 | 2015-04-23T15:26:13.000768 |
+-----+-----+-----+-----+-----+-----+
```

## 9.1.2.8 Data processing service (DPS - Sahara)

### Instal·lació i configuració al node Controller

#### Prerequisites

Abans de realitzar la instal·lació del software pel servei, cal crear la base de dades, les credencials del servei i els punts d'entrada a l'API.

1. Crear la base de dades pel servei **sahara**:
  - a. Connectar-se amb credencials de *root* a la base de dades:

```
santi@Controller:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 5.5.41-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

- b. Crear la base de dades pel DPS:

```
MariaDB [(none)]> CREATE DATABASE sahara;
Query OK, 1 row affected (0.00 sec)
```

- c. Generar les credencials d'accés adequades per la base de dades (usuari: *sahara*, password: *20sahara15*):

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON sahara.* TO 'sahara'@'localhost' \
-> IDENTIFIED BY '20sahara15';
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON sahara.* TO 'sahara'@'%' \
-> IDENTIFIED BY '20sahara15';
Query OK, 0 rows affected (0.00 sec)
```

2. Carregar les variables d'entorn d'admin pel **keystone** per poder fer comandes d'administració (incloses a l'script */home/santi/admin-openrc.sh* que abans he editat):

```
santi@Controller:~$ source admin-openrc.sh
```

3. Crear les credencials del **keystone** pel servei **sahara**:
  - a. Crear l'usuari **sahara**:

```
santi@Controller:~$ keystone user-create \
> --name sahara \
> --pass 20sahara15
```

Property	Value
email	
enabled	True
id	14fd218ed0114c3ab796bc55651a0816
name	sahara
username	sahara

b. Afegir el rol *admin* a l'usuari *sahara* del contenidor de serveis:

```
santi@Controller:~$ keystone user-role-add \
> --user sahara \
> --tenant service \
> --role admin
```

c. Crear l'entitat de servei **sahara**:

```
santi@Controller:~$ keystone service-create \
> --name sahara \
> --type data_processing \
> --description "Data processing service"
```

Property	Value
description	Data processing service
enabled	True
id	823ee238bfae4eaea79b4b6b1420a486
name	sahara
type	data_processing

4. Creació dels punts d'accés a les diferents variants d'APIs (*public*, *internal*, *admin*), es pot veure que es tracta d'urls amb el port corresponent, i de la regió:

```
santi@Controller:~$ keystone endpoint-create \
> --service-id $(keystone service-list | awk '/ sahara / {print $2}') \
> --publicurl http://controller:8386/v1.1/^(tenant_id)s \
> --internalurl http://controller:8386/v1.1/^(tenant_id)s \
> --adminurl http://controller:8386/v1.1/^(tenant_id)s \
> --region regionOne
```

Property	Value
adminurl	http://controller:8386/v1.1/^(tenant_id)s
id	769f8150536949c6ac454f9db7861056
internalurl	http://controller:8386/v1.1/^(tenant_id)s
publicurl	http://controller:8386/v1.1/^(tenant_id)s
region	regionOne
service_id	823ee238bfae4eaea79b4b6b1420a486

## Instal·lar i configurar els components dels servei

Degut a que a la guia s'explica que els paquets no estan disponibles per l'Ubuntu, he hagut de fer recerca per saber com instal·lar el **Sahara**. Hi ha dues opcions:

- Instal·lar els paquets a través del [python-pip<sup>17</sup>](#), l'instal·lador de paquets del Python. D'aquesta manera es faria sobre el sistema operatiu directament, passant a formar part d'aquest. Aquesta manera d'instal·lar la he trobat a diversos posts de diversos fòrums, en concret, [aquest<sup>18</sup>](#) seria el més fàcil de seguir.
- Instal·lar els paquets des d'un entorn *semi-virtual*, creant en primer lloc aquest entorn virtual i després realitzar la instal·lació dels paquets des d'aquest entorn. Aquesta seria la instal·lació que es proposa des de la [documentació oficial específica del Sahara<sup>19</sup>](#).

Per començar a treballar, triaré la opció del fòrum, ja que treballa directament amb el sistema operatiu, sense entorns virtuals. Així, els passos a seguir són:

1. Instal·lar el paquet *python-pip*:

```
santi@Controller:~$ sudo apt-get install python-pip
```

2. Instal·lar els paquets del sahara amb l'instal·lador del python:

```
santi@Controller:~$ sudo pip install sahara
```

3. Comprovar que s'han instal·lat els paquets del sahara:

```
santi@Controller:~$ sudo pip list | grep sahara
python-saharaclient (0.7.3)
sahara (2014.2.1)
```

4. Crear un fitxer de configuració pel **sahara**. Per defecte, a la resta de serveis de l'OpenStack es creava aquest fitxer durant la instal·lació a **/etc/nom\_del\_servei/nom\_del\_servei.conf**. En aquest cas no és així, ni tan sols es crea el directori que l'hauria de contenir. El cas és que a la [documentació específica del servei<sup>20</sup>](#) expliquen que s'ha de crear un fitxer de configuració a partir d'una plantilla anomenada **sahara.conf.sample-basic** (pas 4); doncs bé, després de fer una comanda *find* per trobar el fitxer, obtinc el resultat:

```
santi@Controller:~$ sudo find / -name "sahara.conf.sample-basic"
/usr/local/share/sahara/sahara.conf.sample-basic
```

L'altre qüestió a tenir en compte és que cada cop que s'ha instal·lat un nou servei, s'ha creat un nou usuari anomenat com el servei que representa. Així,

---

<sup>17</sup> <https://docs.python.org/3/installing/>

<sup>18</sup> <http://askubuntu.com/questions/555093/openstack-juno-sahara-data-processing-on-14-04>

<sup>19</sup> <http://docs.openstack.org/developer/sahara/userdoc/installation.guide.html>

<sup>20</sup> <http://docs.openstack.org/developer/sahara/userdoc/installation.guide.html>

cada carpeta que conté els fitxers de configuració té els següents permisos assignats:

```
santi@Controller:~$ ls -lisa /etc/ | grep -e keystone -e glance -e nova -e neutron -e horizon -e heat -e ceilometer
2625609 4 drwxr-xr-x 2 ceilometer ceilometer 4096 Apr 22 18:46 ceilometer
2622968 4 drwxr-x--- 2 glance glance 4096 Apr 23 17:20 glance
2625560 4 drwxr-x--- 3 heat heat 4096 Apr 14 18:18 heat
2622061 4 drwx----- 3 keystone keystone 4096 Feb 23 18:10 keystone
2623063 4 drwxr-x--- 4 root neutron 4096 Mar 3 11:18 neutron
2623033 4 drwxr-x--- 3 nova nova 4096 Mar 5 10:54 nova
2625496 4 drwxr-xr-x 2 horizon horizon 4096 Mar 23 18:13 openstack-dashboard
```

El que ha passat realment ha sigut que s'ha creat un usuari del sistema, pertanyent al grup amb el mateix ID que aquest usuari, sense *home* ni *shell*. Per tant, hauré de fer el mateix en aquest cas pel **sahara**, però manualment:

```
santi@Controller:~$ sudo adduser sahara --system --group --home /var/lib/sahara
Adding system user `sahara' (UID 116) ...
Adding new group `sahara' (GID 125) ...
Adding new user `sahara' (UID 116) with group `sahara' ...
Creating home directory `/var/lib/sahara' ...
```

Un cop tinc l'usuari corresponent, ja puc crear el directori que contindrà el fitxer de configuració, amb els permisos adients. La manera de fer-ho és la següent:

```
santi@Controller:~$ sudo mkdir /etc/sahara -m 750
santi@Controller:~$ sudo chown sahara /etc/sahara/
santi@Controller:~$ sudo chgrp sahara /etc/sahara/
santi@Controller:~$ ls -lisa /etc/ | grep -e keystone -e glance -e nova -e neutron -e horizon -e heat -e ceilometer -e sahara
2625609 4 drwxr-xr-x 2 ceilometer ceilometer 4096 Apr 22 18:46 ceilometer
2622968 4 drwxr-x--- 2 glance glance 4096 Apr 23 17:20 glance
2625560 4 drwxr-x--- 3 heat heat 4096 Apr 14 18:18 heat
2622061 4 drwx----- 3 keystone keystone 4096 Feb 23 18:10 keystone
2623063 4 drwxr-x--- 4 root neutron 4096 Mar 3 11:18 neutron
2623033 4 drwxr-x--- 3 nova nova 4096 Mar 5 10:54 nova
2625496 4 drwxr-xr-x 2 horizon horizon 4096 Mar 23 18:13 openstack-dashboard
2625598 4 drwxr-x--- 2 sahara sahara 4096 Apr 28 17:51 sahara
```

Com es pot observar al requadre verd, el directori s'ha creat correctament. El següent pas és copiar el fitxer *template* al directori corresponent (com a usuari **sahara**):

```
santi@Controller:~$ sudo -u sahara cp /usr/local/share/sahara/sahara.conf.sample \
> /etc/sahara/sahara.conf
```

5. Editar el fitxer de configuració */etc/sahara/sahara.conf* amb el següent contingut:

- a. A la secció [database] cal configurar el paràmetre de connexió a la base de dades (aquesta ha d'existir en aquest punt de la configuració, ja que es crearà posteriorment).

```
# The SQLAlchemy connection string to use to connect to the
# database. (string value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql://sahara:20sahara15@controller/sahara
```

- b. A la secció [keystone\_auth token] s'han de configurar els paràmetres **auth\_uri** on caldrà posar l'API *endpoint* públic del **keystone** pel servei. Llavors al paràmetre **identity\_uri** caldrà posar el d'administració:

```
# Complete public Identity API endpoint (string value)
auth_uri = http://controller:5000/v2.0

# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri = http://controller:35357
```

- c. A la mateixa secció, caldrà especificar l'**admin\_user**, l'**admin\_password** i l'**admin\_tenant\_name**:

```
# Keystone account username (string value)
admin_user = sahara

# Keystone account password (string value)
admin_password = 20sahara15

# Keystone service account tenant name to validate user tokens
# (string value)
admin_tenant_name = service
```

- d. Passant a la secció [DEFAULT], cal configurar els paràmetres de xarxa. S'ha d'especificar que estem utilitzant el **neutron** com a servei de xarxa:

```
# Use Neutron Networking (False indicates the use of Nova
# networking). (boolean value)
use_neutron = true
```

- e. Habilitar el *verbose*:

```
# Print more verbose output (set logging level to INFO instead
# of default WARNING level). (boolean value)
verbose = true
```

6. Com que estem fent servir el DPS (**sahara**) amb una base de dades MariaDB, s'ha de configurar el nombre màxim de paquets permesos per guardar binaris de gran mida de treballs a la base de dades interna del servei. Per fer-ho:

- a. Editar el fitxer de configuració del MySQL a */etc/mysql/my.cnf*, anant a la secció [mysqld] i canviant el paràmetre **max\_allowed\_packet**:

```
#
# * Fine Tuning
#
key_buffer = 16M
max_allowed_packet = 256M
thread_stack = 192K
thread_cache_size = 8
```

b. Reiniciar el mysql:

```
santi@Controller:~$ sudo service mysql restart
* Stopping MariaDB database server mysqld [ OK ]
* Starting MariaDB database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
```

7. Creació de l'esquema de la base de dades:

```
santi@Controller:~$ sudo sahara-db-manage \
> --config-file /etc/sahara/sahara.conf \
> upgrade head
[sudo] password for santi:
INFO [alembic.migration] Context impl MySQLImpl.
INFO [alembic.migration] Will assume non-transactional DDL.
INFO [alembic.migration] Running upgrade None -> 001, Icehouse release
INFO [alembic.migration] Running upgrade 001 -> 002, placeholder
INFO [alembic.migration] Running upgrade 002 -> 003, placeholder
INFO [alembic.migration] Running upgrade 003 -> 004, placeholder
INFO [alembic.migration] Running upgrade 004 -> 005, placeholder
INFO [alembic.migration] Running upgrade 005 -> 006, placeholder
INFO [alembic.migration] Running upgrade 006 -> 007, convert clusters.status_description to LongText
INFO [alembic.migration] Running upgrade 007 -> 008, add security_groups field to node groups
INFO [alembic.migration] Running upgrade 008 -> 009, add rollback info to cluster
INFO [alembic.migration] Running upgrade 009 -> 010, add auto_security_groups flag to node group
INFO [alembic.migration] Running upgrade 010 -> 011, add Sahara settings info to cluster
```

8. Per poder visualitzar l'apartat al Dashboard sobre el servei que s'acaba de muntar (Data Processing), cal tenir-lo arrencat, cosa que es fa de la següent manera:

```
santi@Controller:~$ sudo /usr/local/bin/sahara-all --config-file /etc/sahara/sahara.conf
2015-04-28 21:39:27.162 19159 INFO sahara.main [-] Starting Sahara all-in-one
2015-04-28 21:39:27.164 19159 INFO sahara.utils.rpc [-] Notifications disabled
2015-04-28 21:39:27.499 19159 INFO sahara.plugins.base [-] Plugin 'spark' loaded sahara.plugins.spark.plugin:SparkProvider
2015-04-28 21:39:27.500 19159 INFO sahara.plugins.base [-] Plugin 'hdp' loaded sahara.plugins.hdp.ambariplugin:AmbariPlugin
2015-04-28 21:39:27.501 19159 INFO sahara.plugins.base [-] Plugin 'vanilla' loaded sahara.plugins.vanilla.plugin:VanillaProvider
2015-04-28 21:39:27.540 19159 INFO keystonemiddleware.auth_token [-] Starting keystone auth_token middleware
2015-04-28 21:39:27.542 19159 INFO keystonemiddleware.auth_token [-] Using /tmp/keystone-signing-nCvWUg as cache directory for signing certificate
2015-04-28 21:39:27.543 19159 INFO sahara.main [-] Loading 'all-in-one' ops
2015-04-28 21:39:27.544 19159 INFO sahara.openstack.common.periodic_task [-] Skipping periodic task check_for_zombie_proxy_users because its interval is negative
2015-04-28 21:39:27.545 19159 INFO sahara.main [-] Loading 'direct' infrastructure engine
2015-04-28 21:39:27.558 19159 INFO sahara.main [-] Loading 'ssh' remote
2015-04-28 21:39:27.560 19159 INFO sahara.cli.sahara_all [-] (19159) wsgi starting up on http://0.0.0.0:8386/
```

9. Un cop arrencat, caldrà reiniciar el servei **Apache** (*sudo service apache2 restart*) perquè es puguin visualitzar els canvis al **Dashboard**, fet això es poden veure els resultats:



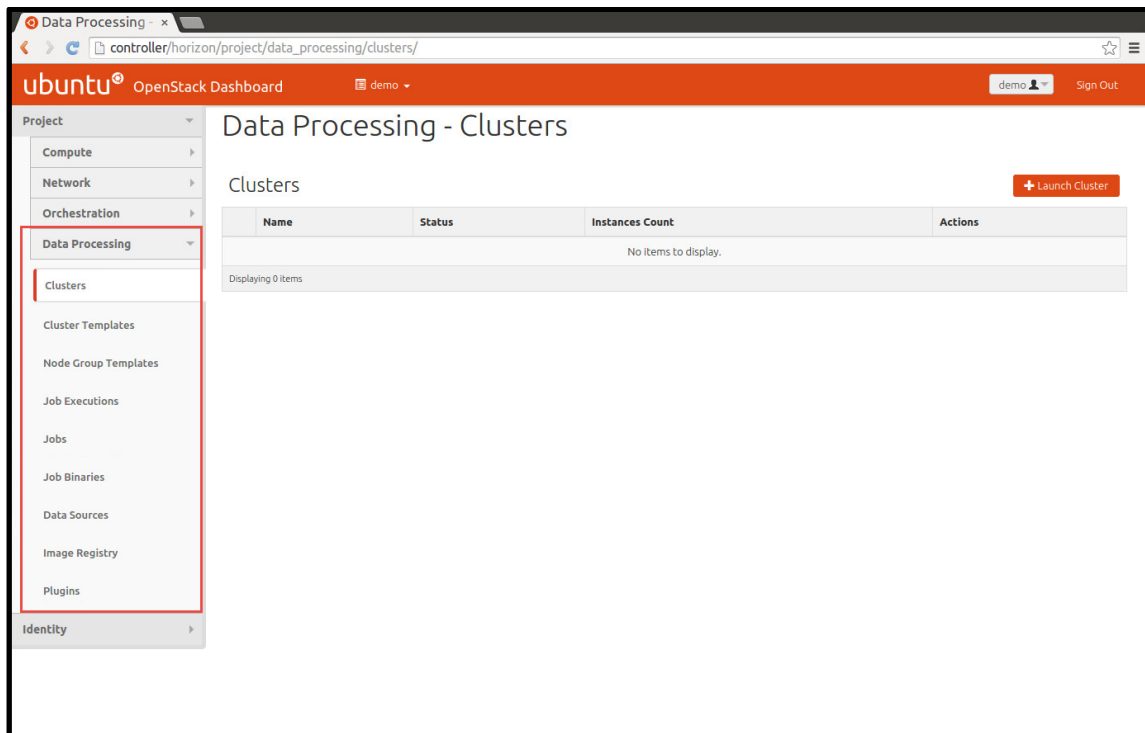


Figura 59: Nous components del Sahara afegits al Dashboard

Com s'ha pogut veure fins ara, en tots els altres serveis que he afegit a l'OpenStack, no he necessitat realitzar cap configuració d'arrencada amb el sistema; a excepció del procés **heat-engine** del servei **Orchestration**, on vaig haver d'afegir una instrucció al seu fitxer d'inici `/etc/init/heat-engine.conf` perquè arrenqués correctament amb el sistema. Doncs bé, com s'ha pogut veure, el *Data Processing Service* (**sahara**) no disposava de paquets d'instal·lació per l'Ubuntu, cosa que m'ha obligat a realitzar una configuració encara més manual. Això ha comportat que tampoc existeixi cap fitxer d'inici que gestioni l'arrencada d'aquest servei, havent-lo d'escriure perquè aquest arrenqui correctament amb el sistema, sent aquest el seu contingut:

```
# vim: set ft=upstart ts=2 et:
description "Sahara Service Engine"
author "Santi Perez <santi.perez.carmona@gmail.com>"

start on runlevel [2345]
stop on runlevel [!2345]

respawn

exec start-stop-daemon \
  --start \
  --chuid sahara \
  --exec /usr/local/bin/sahara-all -- --config-file=/etc/sahara/sahara.conf
```

Aquest fitxer de configuració d'**Upstart** està ubicat a `/etc/init/sahara.conf`.

## Comprovacions

Bé, com a comprovacions, ja n'he fet una, i ha sigut la que he fet al novè pas de la configuració, on he reiniciat el servei Apache i he accedit al Dashboard de l'usuari **demo**, cosa que com s'ha pogut veure, ha funcionat.

Com sempre, també es poden realitzar comprovacions des de línia de comandes, així que les realitzaré des de node **Controller**:

1. Carregar les variables d'entorn per l'usuari **demo**:

```
santi@Controller:~$ source demo-openrc.sh
```

2. Llistar els clústers dels que aquest usuari disposa:

```
santi@Controller:~$ sahara cluster-list
+-----+-----+-----+-----+
| name | id | status | node_count |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

### 9.1.3 Conjunt de serveis

Un cop instal·lats i configurats tots els serveis que s'han explicat al llarg de tot l'apartat 9.2, l'esquema final de serveis i les seves relacions queda reflectit a l'esquema de la següent pàgina, a excepció de:

- *Bare Metal Service*, que només està disponible per la següent versió d'OpenStack (Kilo). Per més claredat s'ha deixat ombrejat.
- *Database Service*, el qual es va decidir no instal·lar perquè pels requeriments del treball no calia.
- *Block Storage*, que tampoc es va muntar pel mateix motiu.
- *Object Storage*, mateix cas que els dos anteriors.

Així doncs, queden configurats definitivament els nodes de l'Openstack per aquest escenari. Ara, el que calia fer era trobar una manera d'extreure els nodes del servidor ESXi, comprovar que aquests funcionen en un entorn semblant al de l'aula i finalment realitzar la implantació a l'aula.

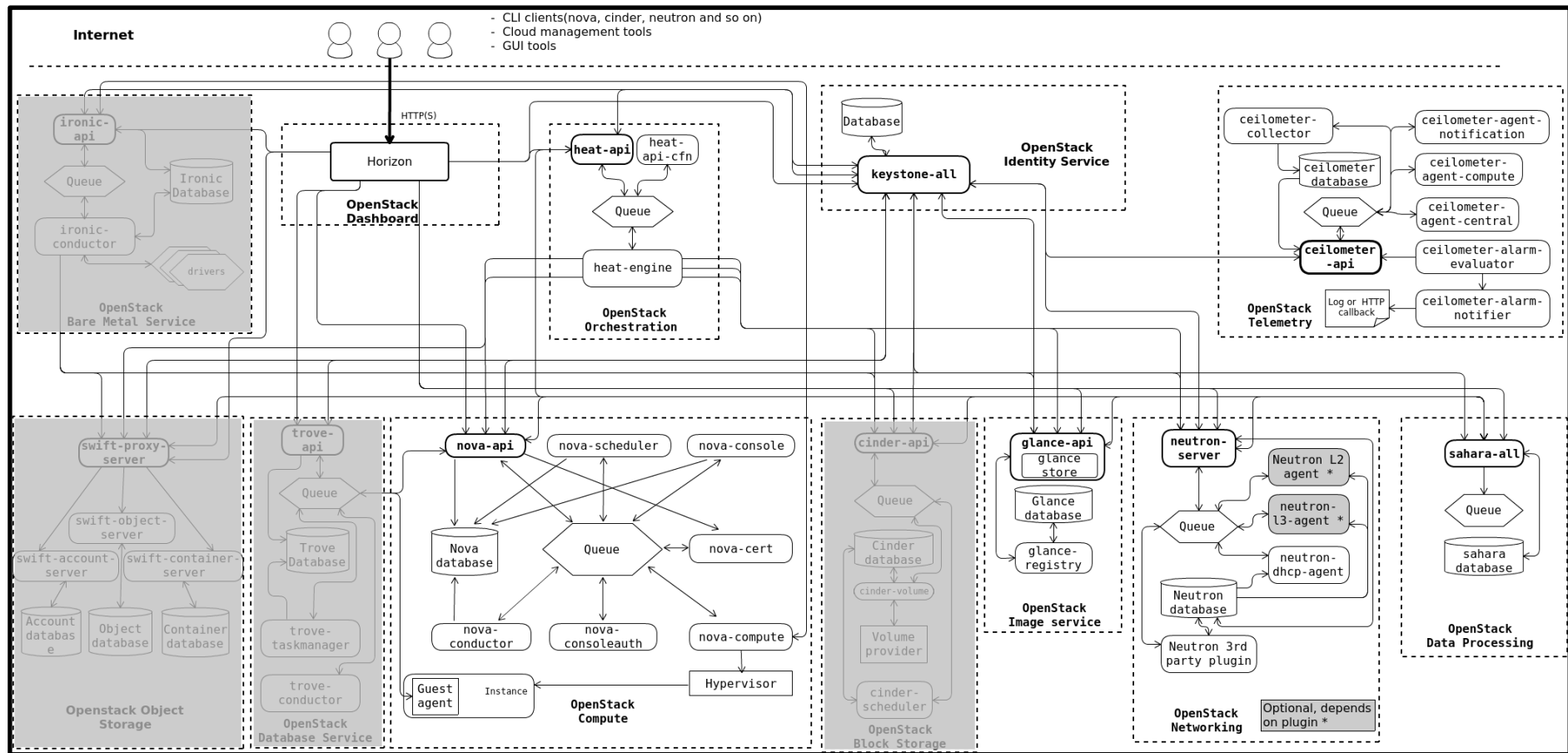


Figura 60: Diagrama dels components que s'han muntat

## 9.2 Escenari PC personal

### 9.2.1 Extracció de les VMs de l'ESXi

El següent pas abans de la implantació a l'aula, era extreure les màquines virtuals que havia configurat fins al moment a l'ESXi. Vaig fer una mica de recerca a la web i vaig trobar que VMware disposa d'una eina anomenada *VMware Converter* per tal fi. Només cal instal·lar aquesta eina i seguir els passos que es descriuran a continuació.

1. Arrencar l'assistent de conversió de màquines:

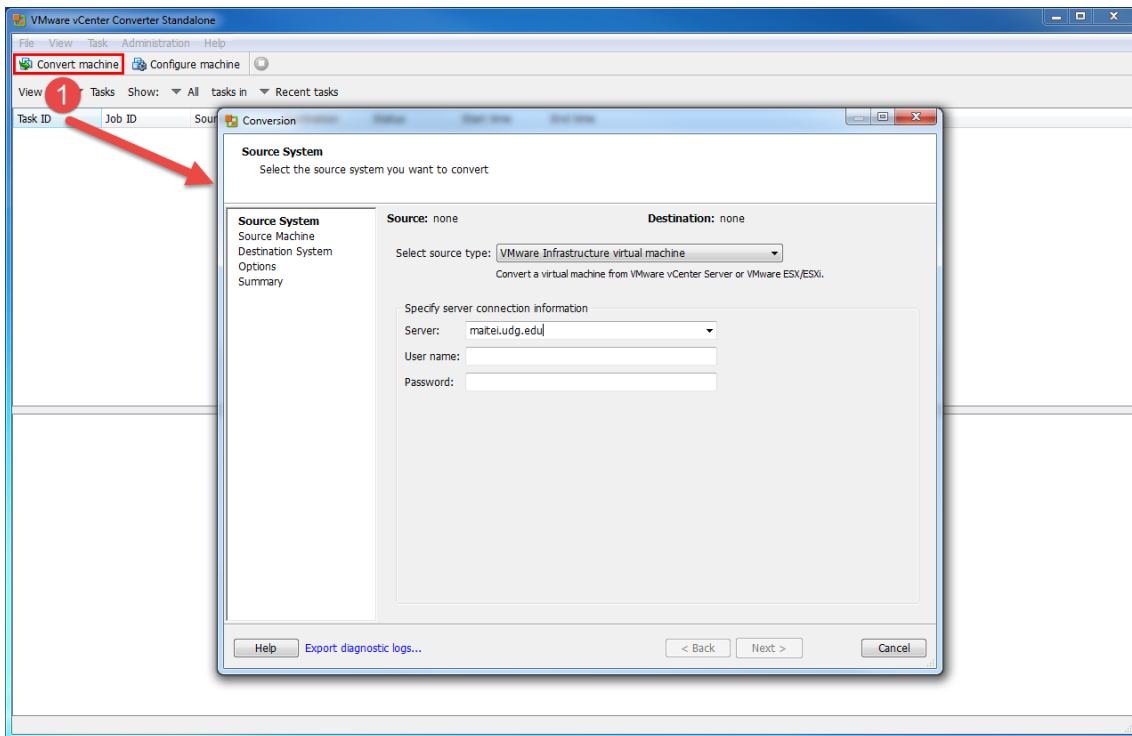


Figura 61: Primer pas per l'extracció

2. Triar el tipus de font i completar la informació per connectar-se al servidor:

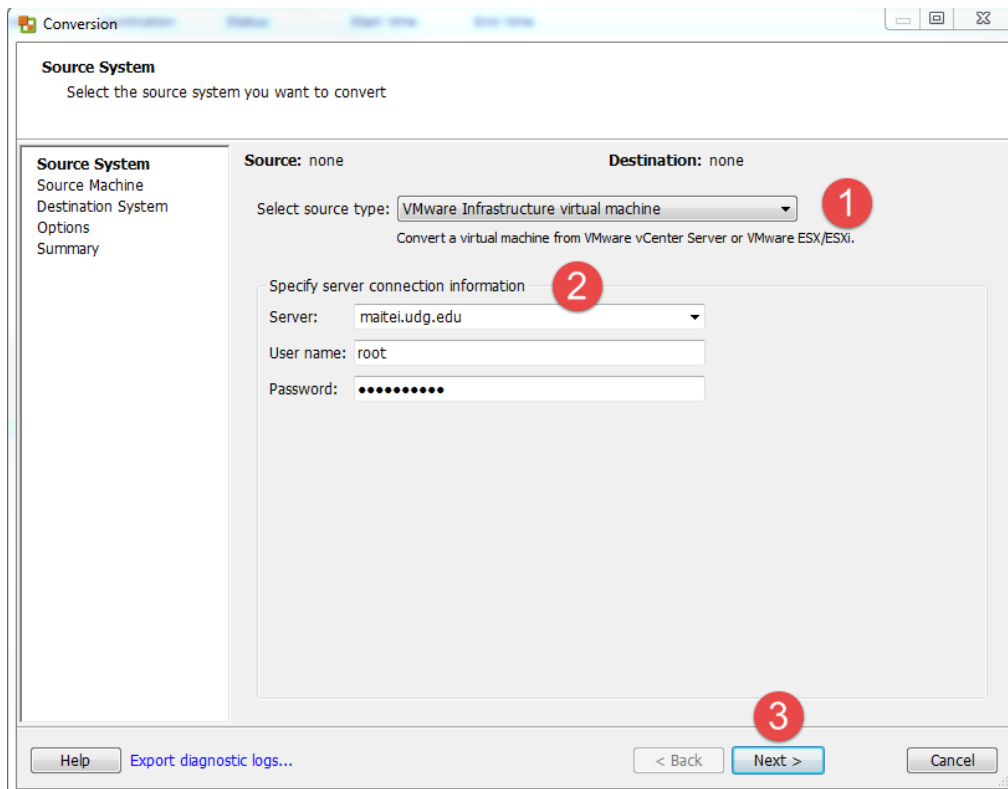


Figura 62: Introducció de credencials

3. Un cop connectats, el servidor ens mostrarà les màquines virtuals de que disposa; caldrà triar-ne una i anar al pas següent. IMPORTANT! Cal que la màquina virtual estigui aturada, en cas contrari, es produirà un error:

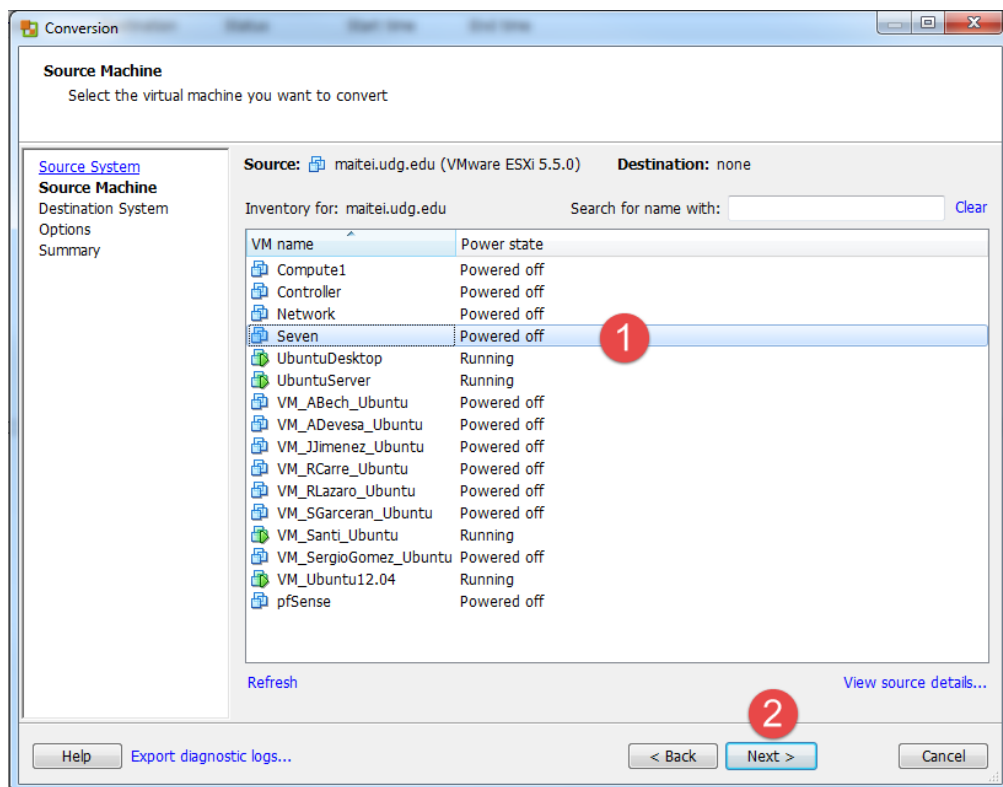


Figura 63: Selecció de la màquina virtual per extreure

4. S'arrencarà un procés en el qual s'obindrà tota la informació requerida sobre la VM. Aquest pot trigar una estoneta (5 minuts més o menys) en funció dels recursos i la mida:

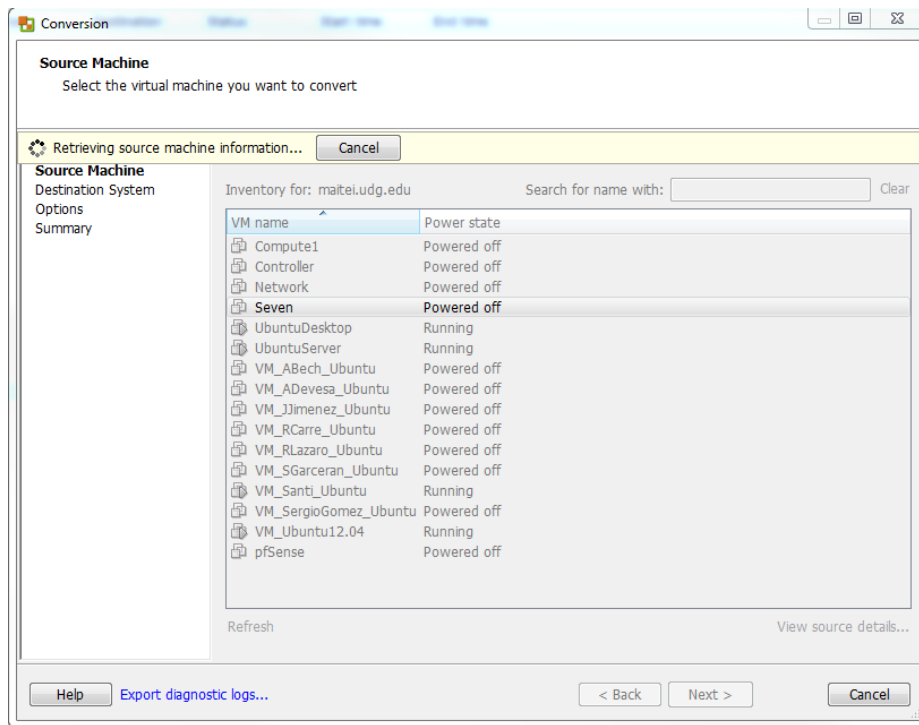


Figura 64: Post-selecció

5. El següent pas es tracta de triar el tipus de VM i producte VMware de destí, posar-li un nom i indicar on es vol guardar:

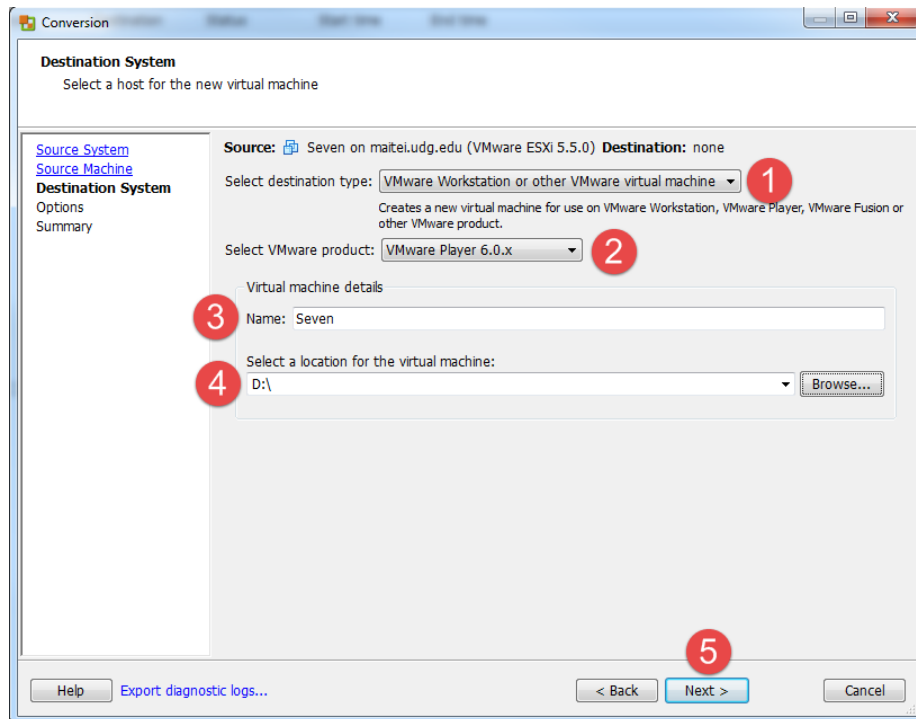


Figura 65: Introducció de les dades de sortida

6. A continuació es mostra un resum dels recursos de la VM, amb els possibles conflictes que tindria exportar-la a la màquina destí. En aquest cas avisa que no té prou espai de disc (cosa no del tot certa, ja que aquesta VM té un disc *Thin provisioned*, és a dir, només ocupa el que va necessitant). Per canviar algun paràmetre de destí, només cal prémer el botó *Edit* corresponent i fer-ho, tot i que encara que no es faci en aquest punt, es podrà modificar des de l'VMware Player:

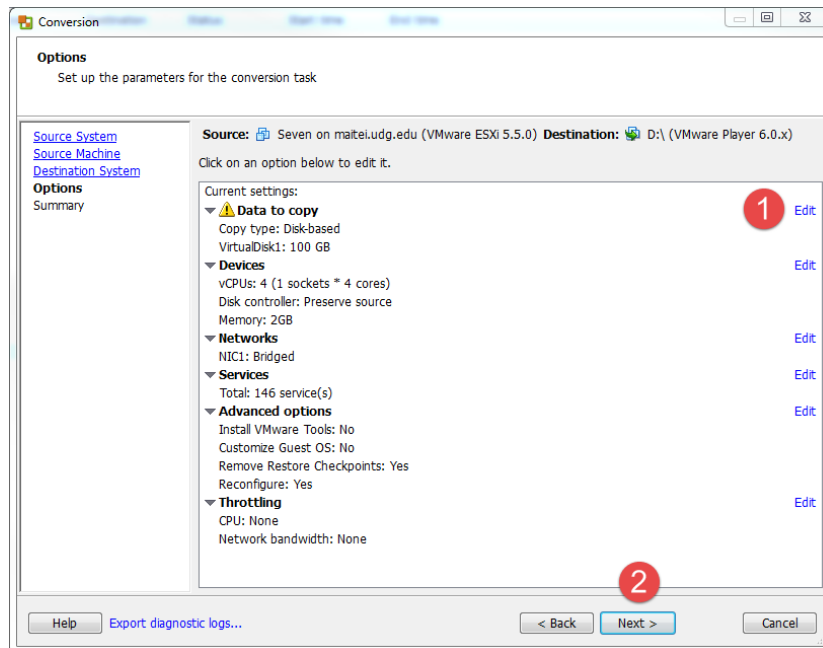


Figura 66: Resum de conflictes

7. Al pas final ens mostra el resum de les accions que s'han decidit en transcurs de l'assistent, avisant també, de qualsevol possible conflicte:

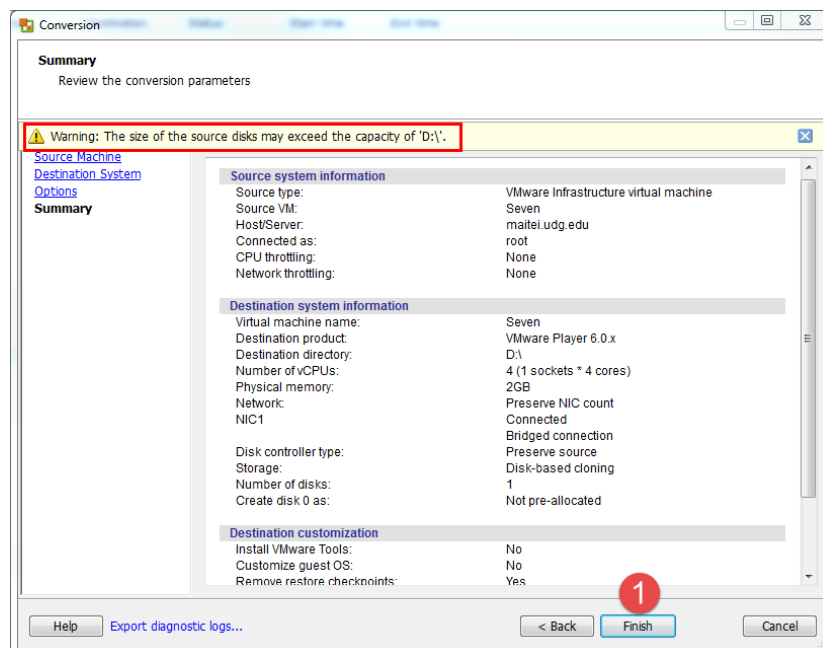


Figura 67: Resum de les accions a realitzar

8. Finalment podrem veure a la finestra principal de la utilitat una nova tasca corresponent a l'assistent acabat de seguir:

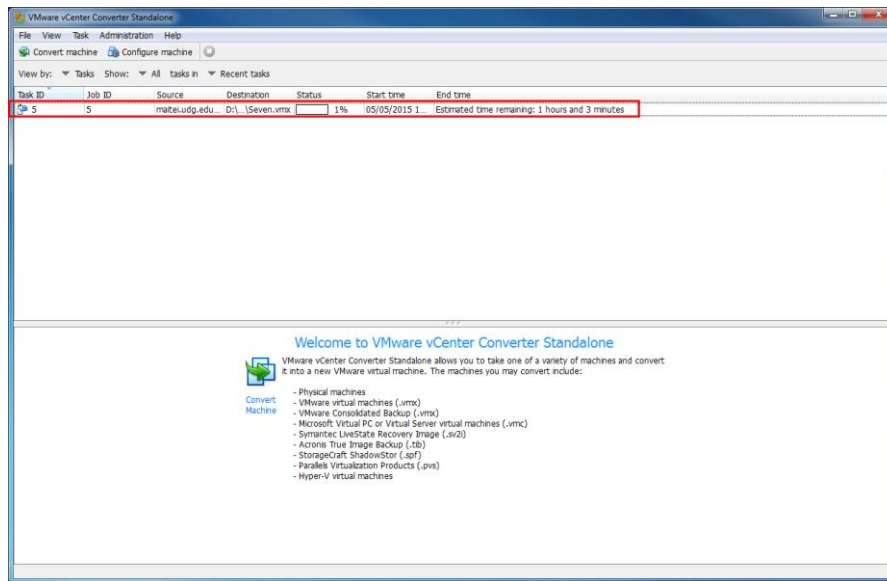


Figura 68: Consola principal del VMware Converter

## 9.2.2 Procés d'adaptació a l'escenari

Tal i com s'havia explicat a l'apartat d'estudis i decisions, la plataforma de virtualització que triaria per l'aula seria l'VMware Player, la mateixa que provaria amb el PC personal. Així, en primer lloc calia familiaritzar-se amb ella i comprovar que el cloud seguia funcionant tal i com ho feia a l'ESXi.

Degut a la inferioritat de recursos de la màquina personal, calia fer una reducció dels recursos que necessitaven els nodes de l'OpenStack per a poder realitzar una execució correcta.

### Disminuir la memòria disponible

Aquesta configuració era realment senzilla, i es tractava de:

1. Seleccionar la VM i editar-ne les característiques:



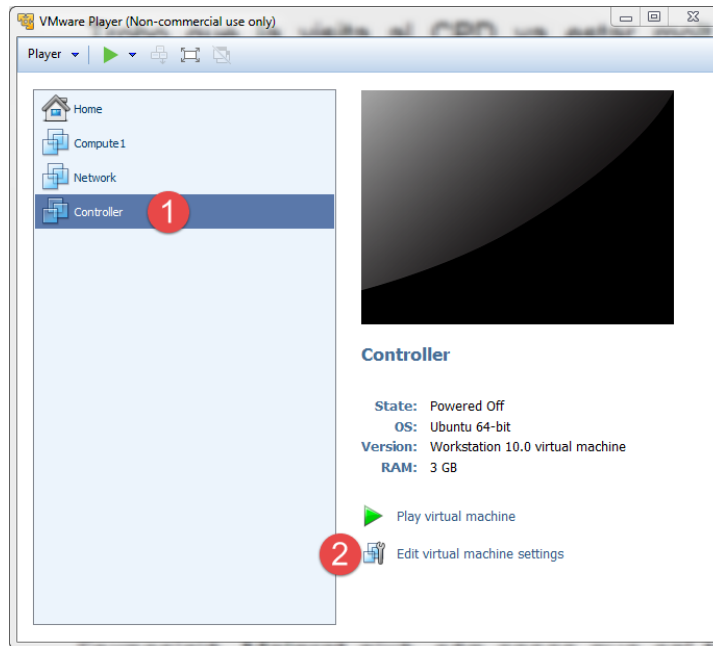


Figura 69: Finestra principal de l'VMware Player

2. Escriure la nova mida del recurs i confirmar:

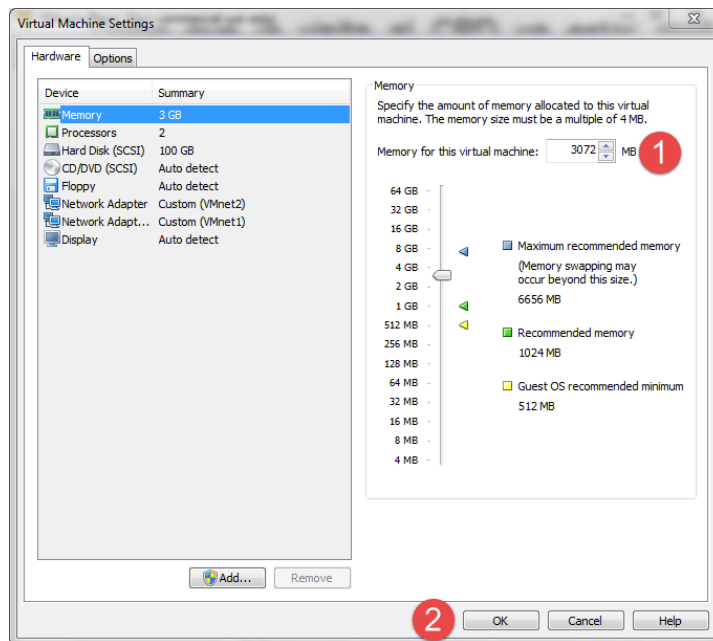


Figura 70: Propietats de la màquina virtual

Com ja he explicat anteriorment, la “nova” màquina disposava de menys RAM, concretament de 8GB. Així, per poder fer córrer el cloud en un sistema estable, la repartició d'aquest recurs va ser la següent:

- 3GB pel node *Controller*.
- 3GB pel node *Compute1*.
- 1GB pel node *Network*.

## Creació de les xarxes Management, Tunnel i External

Les primeres xarxes que vaig crear van ser les internes del cloud, és a dir, *Management* i *Tunnel*. El problema és que el *Player*, al ser la versió més senzilla, només ofereix tres modes de xarxa: *bridged*, *NAT* i *Host-only*. Doncs bé, el mode de treball per aquestes LANs que es necessitava per aquest escenari seria com el mode *internal* del VirtualBox, mode en qual la LAN és d'àmbit de les VMs i prou. De fet, a la versió 6 del *Player*, aquest mode existia, però amb la nova versió 7.1, l'han tret.

Existeixen dues utilitats que permeten afegir i administrar xarxes a l'VMware:

- **vnetlib**. Es tracta d'un executable ubicat al directori d'instal·lació del *Player* (*C:\Program Files (x86)\VMware\VMware Player*) que permet no només afegir i administrar xarxes, sinó tot el maquinari virtual de que disposa el *Player*. Per fer-ne ús, cal fer-ho a través de línia de comandes (anant al *path* abans esmentat i invocant-lo amb els paràmetres corresponents). [Aquí<sup>21</sup>](#) es pot consultar un resum de les possibles accions realitzables amb aquesta eina.
- **vmnetcfg**. També és un executable i correspon a l'eina específica de creació i administració de xarxes de l'VMware, amb interfície gràfica. El que fa realment aquesta eina és treballar per sobre del *vnetlib*, és a dir, totes les configuracions realitzades s'acaben traduint en comandes d'*vnetlib*.

Com és d'esperar, l'eina més amigable és l'*vmnetcfg*, però hi ha un problema, i és que només està disponible per al *Workstation* i no per al *Player*, bé, en teoria. Com expliquen en [aquesta web<sup>22</sup>](#), aquesta eina es pot treure dels fitxers temporals d'instal·lació del *Workstation*. El procés que jo vaig seguir va ser el següent:

1. Extreure manualment els fitxers de l'instal·lador a un directori:

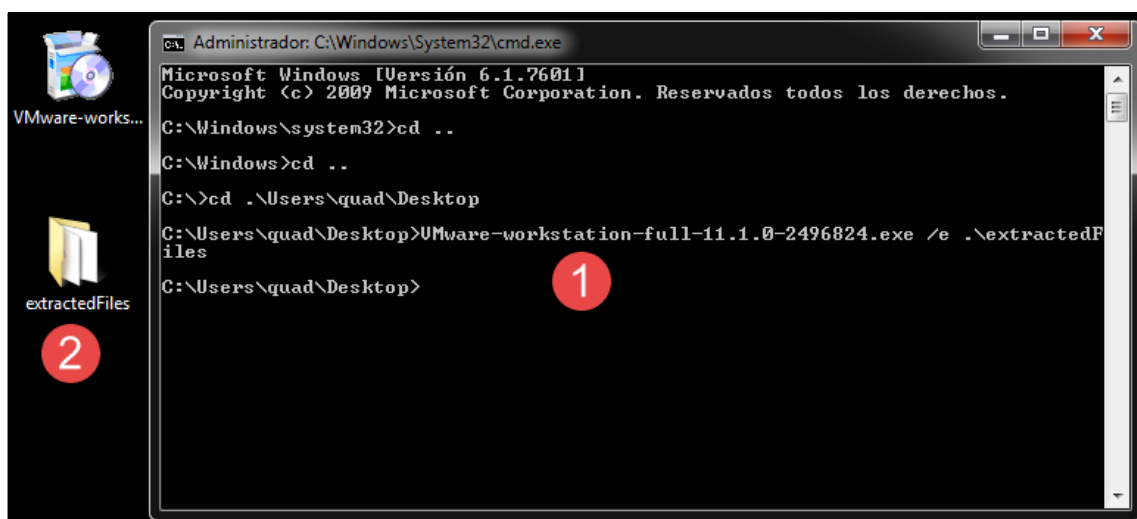


Figura 71: Procés d'extracció

<sup>21</sup> <http://sanbarrow.com/network/cmdguide2workstation.html>

<sup>22</sup> <http://www.eightforums.com/virtualization/5137-how-add-virtual-network-editor-vmware-player.html>

2. Buscar el fitxer d'instal·lació de Windows (*nom.msi*) i obrir-lo amb el programa 7zip:

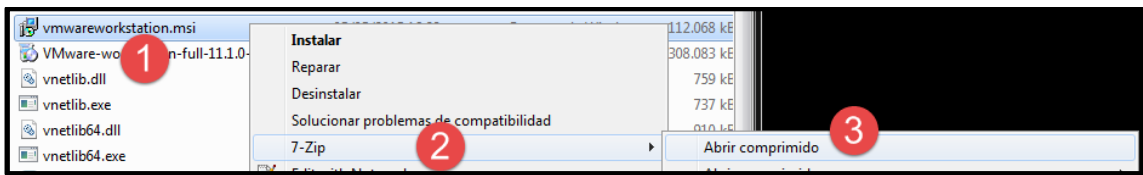


Figura 72: Instruccions d'accés al fitxer

3. Buscar-hi el fitxer "core.cab" i obrir-lo també:

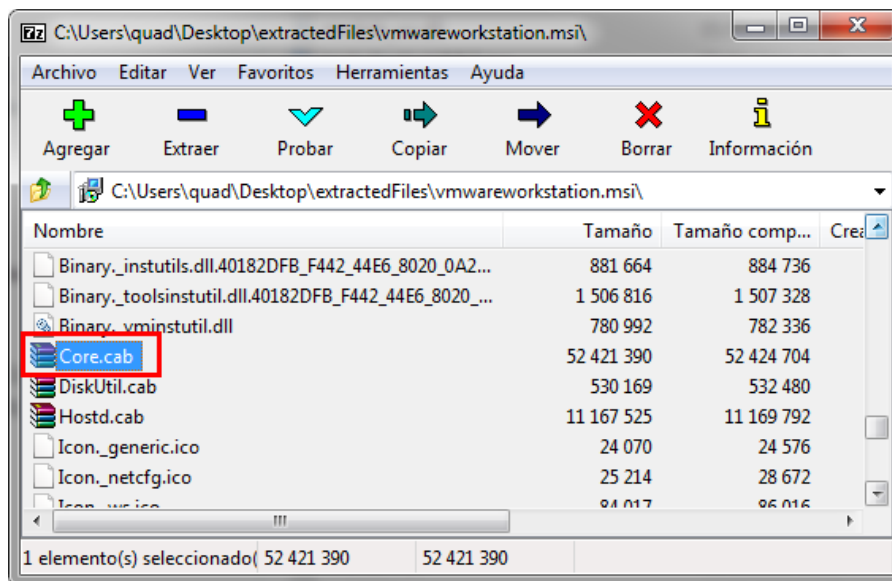


Figura 73: Cerca del fitxer core.cab

4. Buscar l'eina (*vmnetcfg*) i copiar-la al directori d'instal·lació del *Player*:

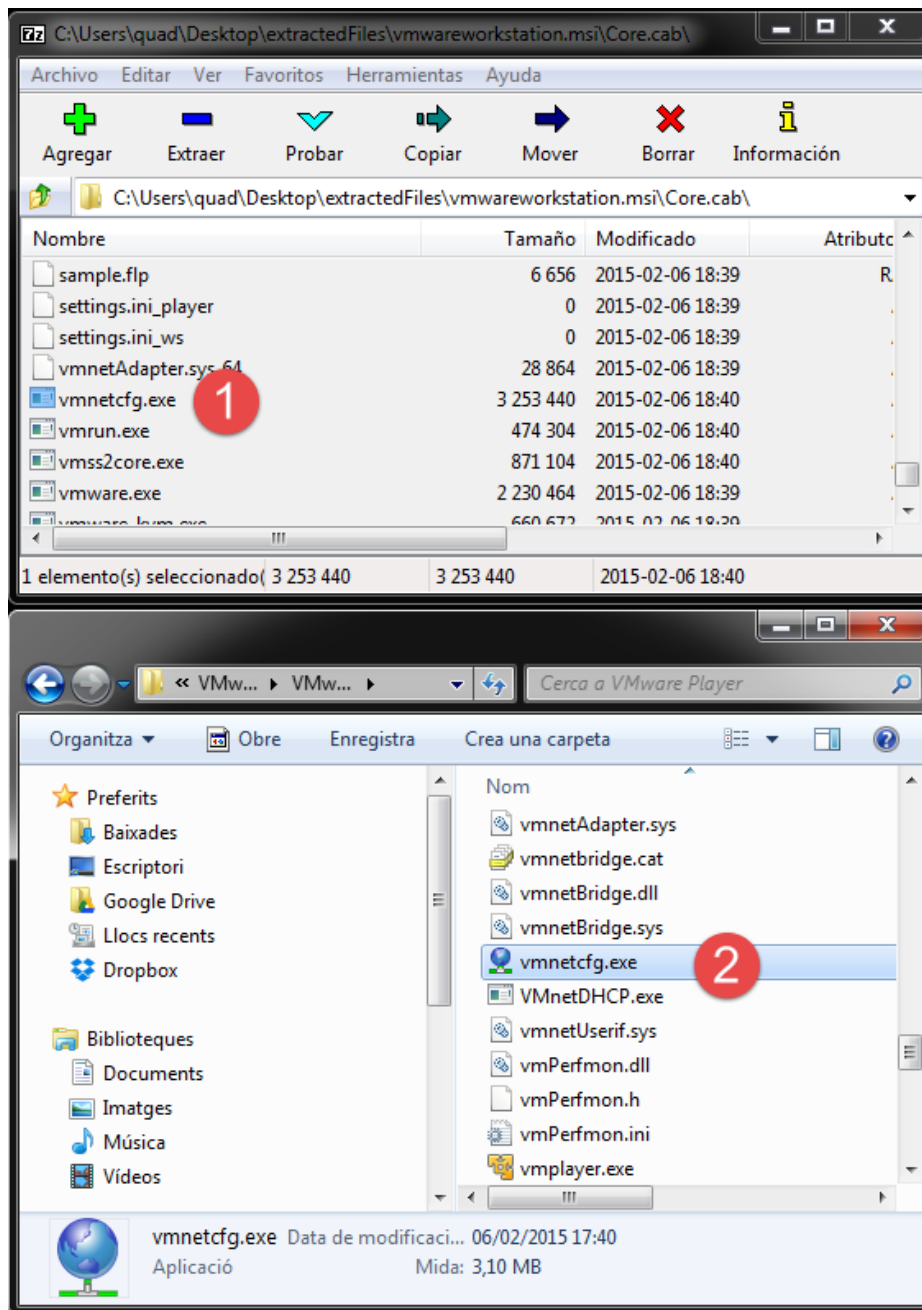


Figura 74: Còpia de l'eina al path corresponent

5. Executar-la com a administrador per poder fer-la servir:

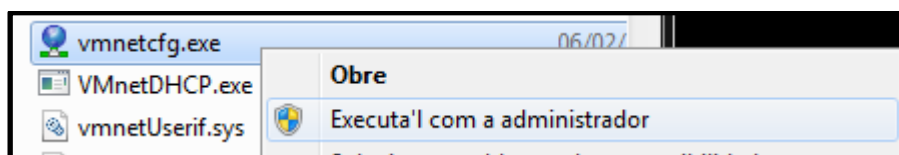


Figura 75: Execució com administrador

6. Interfície del vmnetcfg:

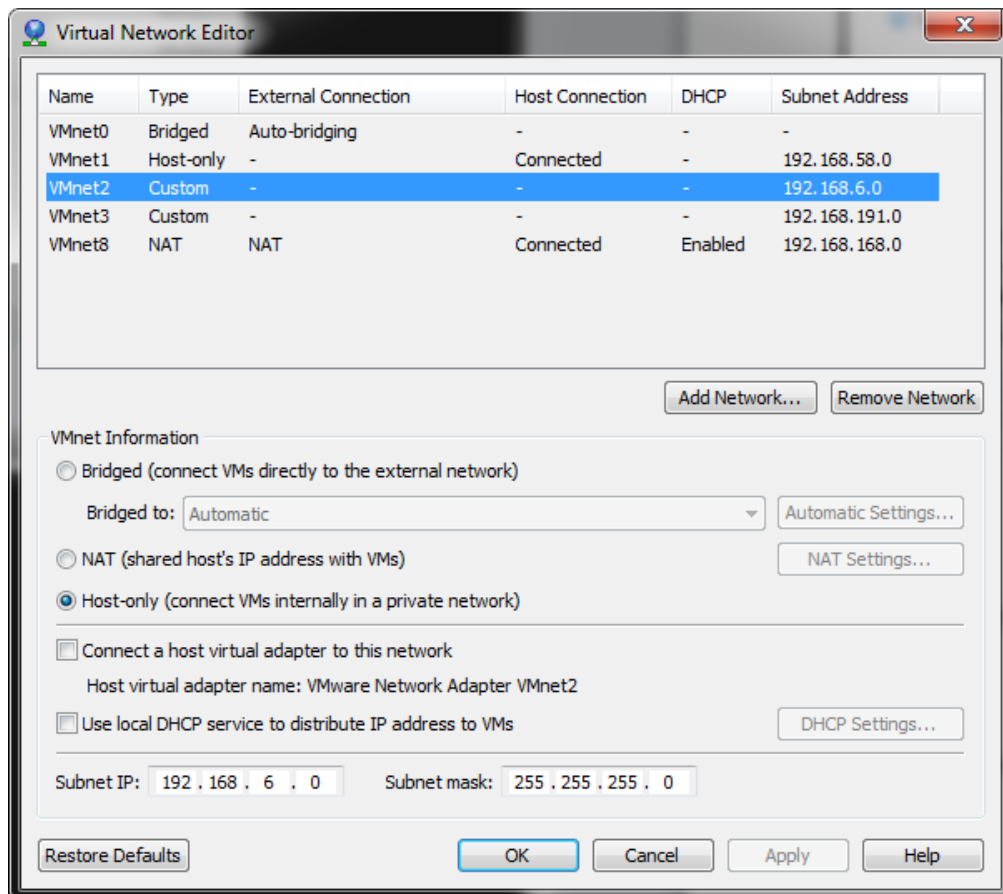


Figura 76: Interfície de l'eina

La interfície que aquí mostro es correspon a després d'haver afegit les xarxes que necessitava (VMnet2 i VMnet3). Com es pot veure, està seleccionada VMnet2, els paràmetres que li corresponen són els que es poden veure a la mateixa interfície. Simplement per afegir-la, cal prémer el botó "Add Network..." i posar els paràmetres adients. Les correspondències amb les xarxes de l'OpenStack són les següents:

- *Management* = VMnet2
- *Tunnel* = VMnet3
- *External* = VMnet0

Així, segons les equivalències que acabo de proposar, l'esquema de l'escenari és el següent:

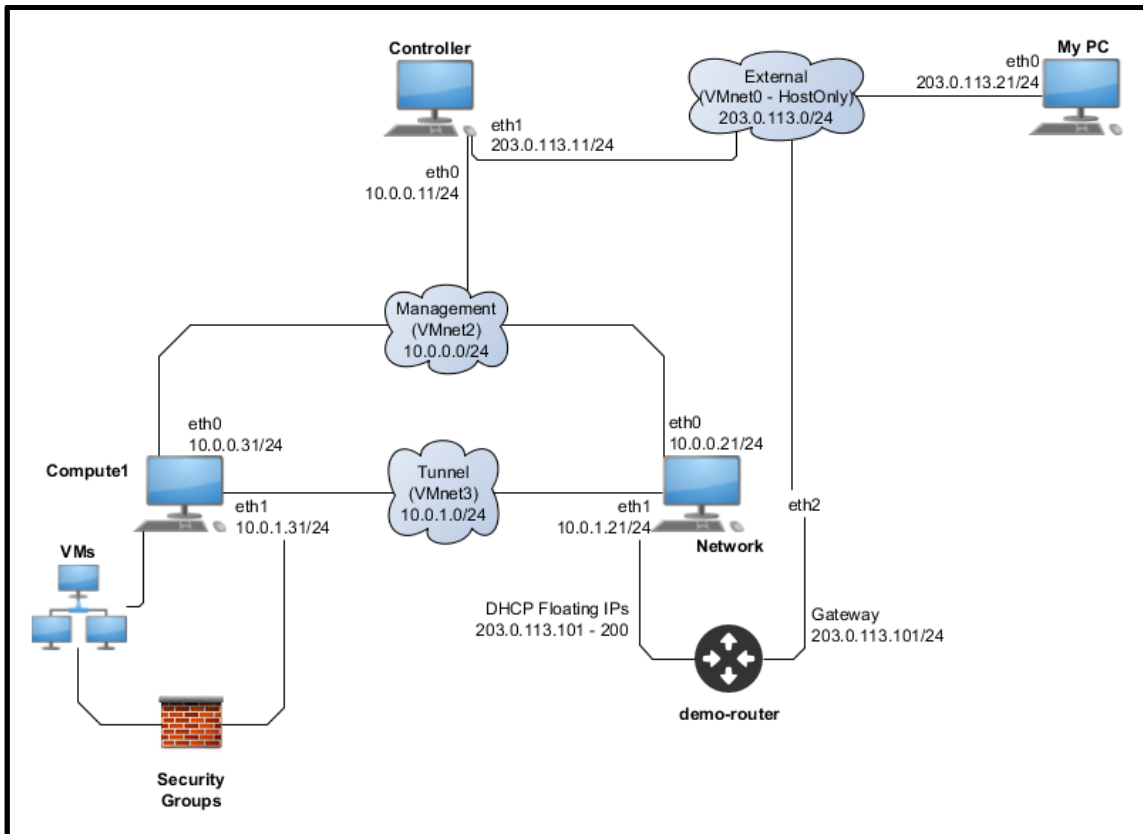


Figura 77: Escenari a casa

Un cop vaig tenir l'esquema de l'escenari clar, només va caler configurar adientment les màquines:

- *Controller*:

Memory	3 GB
Processors	2
Hard Disk (SCSI)	100 GB
CD/DVD (SCSI)	Auto detect
Floppy	Auto detect
Network Adapter	Custom (VMnet2)
Network Adapt...	Custom (VMnet1)
Display	Auto detect

Figura 78: Especificacions Controller

- *Compute1*:

Memory	3 GB
Processors	4
Hard Disk (SCSI)	120 GB
CD/DVD (SCSI)	Auto detect
Floppy	Auto detect
Network Adapter	Custom (VMnet2)
Network Adapt...	Custom (VMnet3)
Display	Auto detect

Figura 79: Especificacions Compute1

- Network:

Memory	1 GB
Processors	2
Hard Disk (SCSI)	50 GB
CD/DVD (SCSI)	Auto detect
Floppy	Auto detect
Network Adapter	Custom (VMnet2)
Network Adapt...	Custom (VMnet3)
Network Adapt...	Custom (VMnet1)
Display	Auto detect

Figura 80: Especificacions Network

## Verificacions

Un cop configurades les màquines adientment, vaig seguir els següents passos:

- Arrencar els nodes en l'ordre correcte com sempre (*Controller*, *Network* i *Compute1*).
- Comprovar la connectivitat entre les interfícies:

```
santi@Controller:~$ ping -c 4 10.0.0.21
PING 10.0.0.21 (10.0.0.21) 56(84) bytes of data.
64 bytes from 10.0.0.21: icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from 10.0.0.21: icmp_seq=2 ttl=64 time=0.293 ms
64 bytes from 10.0.0.21: icmp_seq=3 ttl=64 time=0.304 ms
64 bytes from 10.0.0.21: icmp_seq=4 ttl=64 time=0.241 ms

--- 10.0.0.21 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.241/0.277/0.304/0.031 ms
santi@Controller:~$ ping -c 4 10.0.0.31
PING 10.0.0.31 (10.0.0.31) 56(84) bytes of data.
64 bytes from 10.0.0.31: icmp_seq=1 ttl=64 time=0.460 ms
64 bytes from 10.0.0.31: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from 10.0.0.31: icmp_seq=3 ttl=64 time=0.361 ms
64 bytes from 10.0.0.31: icmp_seq=4 ttl=64 time=0.288 ms

--- 10.0.0.31 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.270/0.344/0.460/0.078 ms
```

```
santi@Controller:~$ ping -c 4 203.0.113.21
PING 203.0.113.21 (203.0.113.21) 56(84) bytes of data.
64 bytes from 203.0.113.21: icmp_seq=1 ttl=128 time=0.481 ms
64 bytes from 203.0.113.21: icmp_seq=2 ttl=128 time=0.389 ms
64 bytes from 203.0.113.21: icmp_seq=3 ttl=128 time=0.255 ms
64 bytes from 203.0.113.21: icmp_seq=4 ttl=128 time=0.258 ms

--- 203.0.113.21 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.255/0.345/0.481/0.097 ms
```

```
santi@Network:~$ ping -c 4 10.0.1.31
PING 10.0.1.31 (10.0.1.31) 56(84) bytes of data.
64 bytes from 10.0.1.31: icmp_seq=1 ttl=64 time=0.313 ms
64 bytes from 10.0.1.31: icmp_seq=2 ttl=64 time=0.278 ms
64 bytes from 10.0.1.31: icmp_seq=3 ttl=64 time=0.331 ms
64 bytes from 10.0.1.31: icmp_seq=4 ttl=64 time=0.314 ms

--- 10.0.1.31 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.278/0.309/0.331/0.019 ms
```

```
C:\Users\quad>ping 203.0.113.101

Haciendo ping a 203.0.113.101 con 32 bytes de datos:
Respuesta desde 203.0.113.101: bytes=32 tiempo<1m TTL=64
Respuesta desde 203.0.113.101: bytes=32 tiempo<1m TTL=64
Respuesta desde 203.0.113.101: bytes=32 tiempo<1m TTL=64
Respuesta desde 203.0.113.101: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 203.0.113.101:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

- Afegir el nom **controller** al fitxer hosts de l'ordinador (pc de casa):

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com             # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1          localhost
#   ::1                localhost
203.0.113.11         controller
```

- Accedir al *Dashboard* amb el navegador:



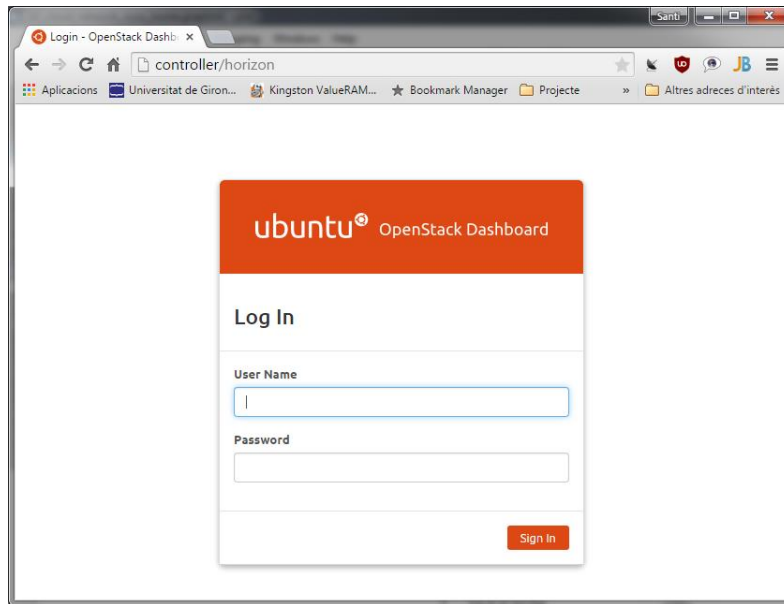


Figura 81: Dashboard de l'OpenStack a casa

- Arrencar una instància visualitzar-la amb el VNC i parar-la:

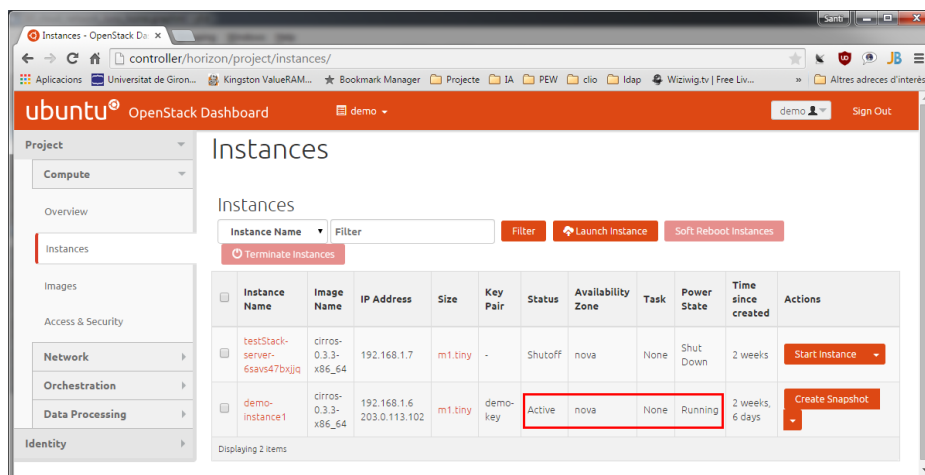


Figura 82: Engedada d'una instància

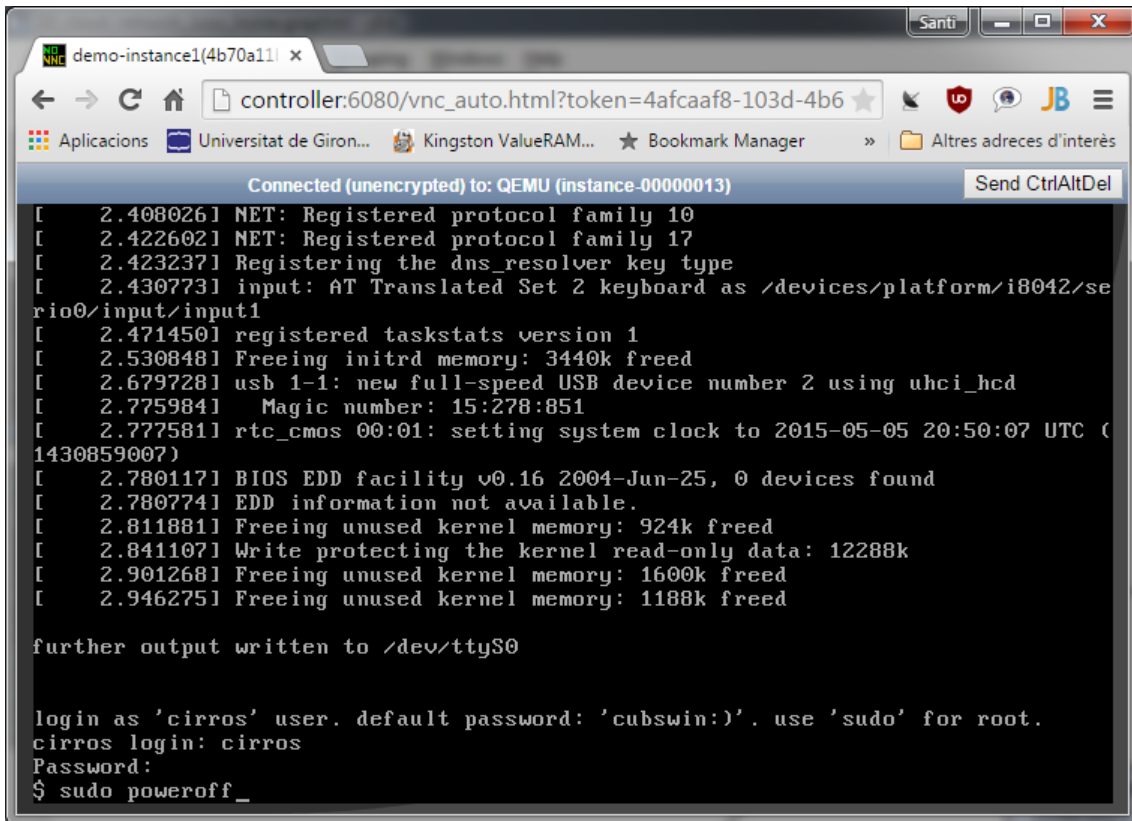


Figura 83: Consola de la instància

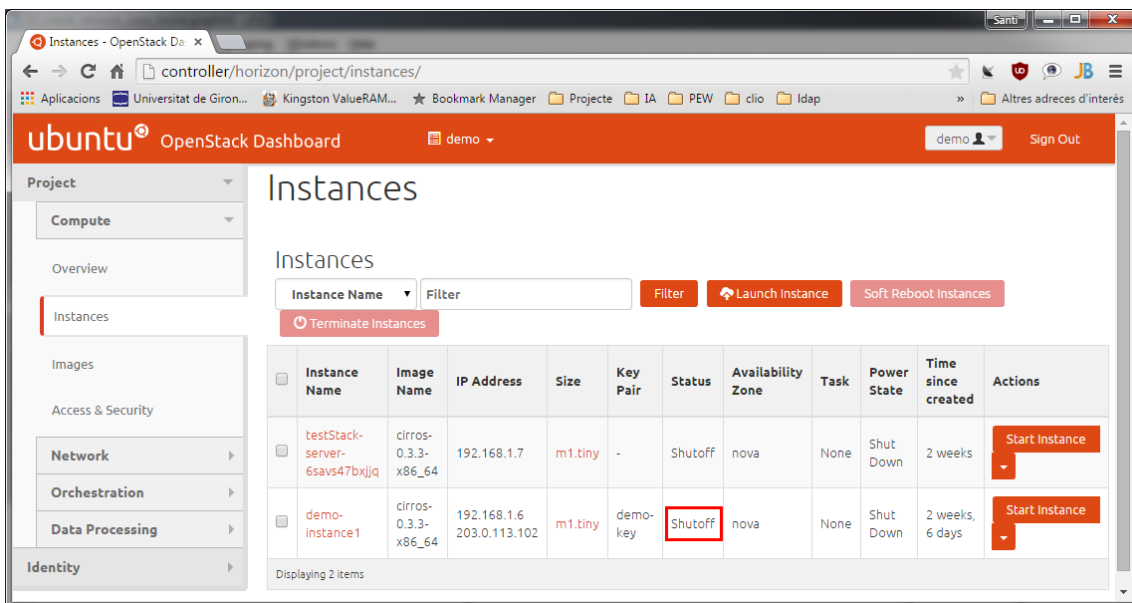


Figura 84: Aturada de la instància

Doncs bé, com es pot comprovar, tot funcionava correctament.

### 9.3 Escenari final: Aula III-04i

Finalment, aquesta és la darrera part del treball pel que fa a preparacions respecte a l'execució del *cloud* a l'aula i la seva automatització. D'una banda s'ha preparat un servidor Windows Server 2003 virtualitzat amb VMware amb una sèrie d'eines que permeten administrar l'aula tan automàticament com activament, tot de manera remota. D'altra banda s'han preparat les 18 estacions de l'aula per poder executar un *cloud* OpenStack, de manera independent a la docència.

En els apartats que segueixen s'explicarà en primer lloc com s'ha implementat el servidor i les eines configurades a més de les desenvolupades. Seguidament es parlarà del procés de configuració i adaptació de les estacions per desenvolupar les noves tasques.

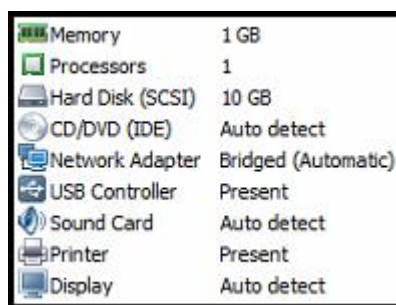
#### 9.3.1 Servidor d'aula

Al llarg de la memòria s'ha anat explicant que s'instal·laria la màquina que tenia al laboratori de BCDS per allotjar un servidor que permetés el control administratiu de l'aula en les seves dues funcions: *cloud* i aula docent. A partir d'ara, em referiré a aquesta màquina com a Comunicacions19, el seu nom de *host*.

#### Tasques i eines

La primera tasca que vaig realitzar va ser la d'instal·lar la darrera versió d'*VMware Player* a Comunicacions19 per després crear una nova màquina que tingués com a sistema operatiu el Windows Server 2003 Standard Edition, amb no gaires recursos, donat que es tracta d'un sistema operatiu que no els requereix i que a més, per les tasques que ha de desenvolupar tampoc calen. Així, els primers passos que es van realitzar van ser:

1. Creació d'una VM sense gaires recursos:



Memory	1 GB
Processors	1
Hard Disk (SCSI)	10 GB
CD/DVD (IDE)	Auto detect
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Figura 85: Recursos del Windows Server 2003

2. Instal·lació del Windows 2003 Server Standard Edition, instal·lar antivirus (ClamAV), instal·lar el darrer *Service Pack* disponible (SP2). Configuració de xarxa (descrita a l'apartat 8.3, Escenari final)
3. Instal·lar el programa d'administració del *Deepfreeze*, que inclou la consola (*Deepfreeze console*) que permet administrar les estacions que tenen instal·lat el *Deepfreeze* client, i també el *Deepfreeze Administrator* que permet crear nous programes de congelació i llavors:

- a. Els programes de congelació són els que s'instal·len a les màquines que es desitja congelar i administrar.
- b. Les llavors permeten administrar també les màquines des de la consola, però no les congela, podent instal·lar el programa de congelació posteriorment des de la pròpia consola.

Tot i trobar-me les estacions de l'aula amb el *Deepfreeze* instal·lat, encara que vaig instal·lar la consola al servidor, aquest no era capaç de veure-les. El motiu és que la consola només pot administrar màquines a les quals se'ls hagi instal·lat un programa de congelació que s'hagi generat amb dita consola (bé, el seu component *Administrator*). Al no ser el cas, vaig haver de descongelar l'aula màquina per màquina i instal·lar el nou "programet" de congelació que vaig crear per aquest fi, per finalment poder visualitzar i administrar les màquines des de la consola:

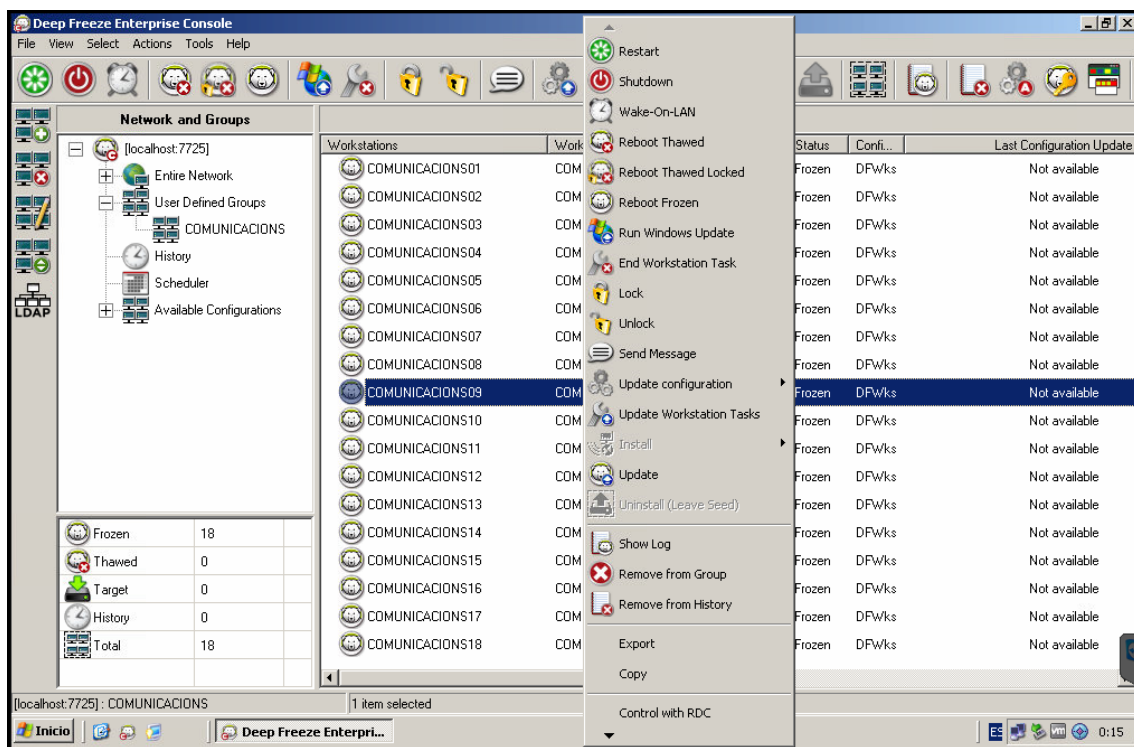


Figura 86: Consola Deepfreeze

Com es pot observar a la figura anterior, totes les màquines de l'aula apareixen a la consola del *Deepfreeze*, i es poden fer totes les tasques del menú contextual per les que es vulguin, és a dir, per les que es seleccionin per realitzar la tasca concreta. Per posar un exemple, es poden reiniciar totes les màquines descongelades alhora.

4. Disposa d'un recurs compartit al qual hi té accés un dels usuaris (OpenStack) present a les estacions, que també és en d'aquest servidor. D'aquesta manera, si les estacions han de disposar d'alguna dada o executar un *batch file* allotjat al servidor, ho poden fer.

5. Vaig programar uns quants *batch files* que permeten realitzar tasques administratives a les estacions, els quals es poden trobar als annexos.
6. Instal·lació del *WakeOnLan*. Es tracta d'una eina molt útil, la seva principal característica és que permet arrencar les màquines a través de la xarxa, però si a més es configura gràficament, ofereix una visió de les màquines que es troben arrencades, si hi ha connectivitat entre el servidor i aquestes, control remot... La configuració gràfica s'aconsegueix fàcilment editant un fitxer de configuració xml.

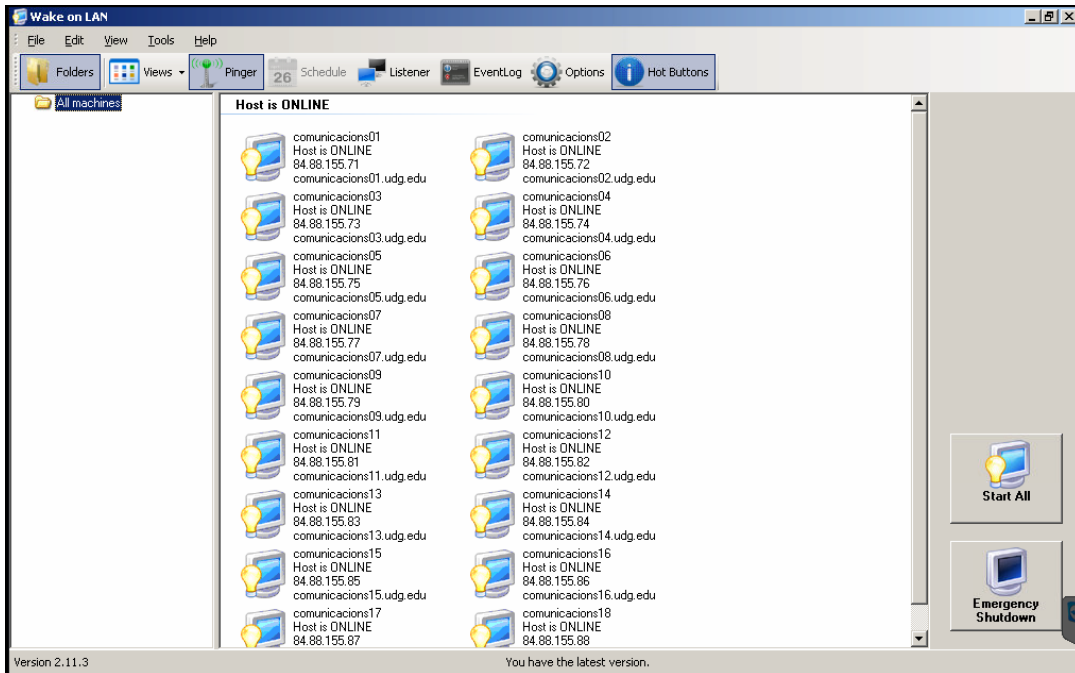


Figura 87: Consola WakeOnLan

## Cloud Control

Finalment, la darrera tasca, i la que m'ha portat més temps al servidor ha estat l'automatització de l'arrencada i la parada del *cloud*. Els principals problemes que m'he trobat per poder realitzar aquestes tasques, han estat els permisos del Windows a les estacions. Amb el control de comptes habilitat (UAC) i a més a més que l'usuari OpenStack no sigui administrador (com es veurà a la descripció de les tasques realitzades a les estacions), provoca conflictes per poder executar comandes i programes remotament.

Tot i els problemes, he aconseguit que funcioni, però la veritat és molt millorable, tal i com quedarà reflectit a l'apartat de Treball futur. Passem a explicar el *batch file*; en primer lloc tenim la definició d'una sèrie de variables que necessita per treballar:

```

SET hostsFile="hosts.txt"
SET computeNodes="computeNodes.txt"
SET networkNode="networkNode.txt"
SET controllerNode="controllerNode.txt"
SET wakeOnLan="C:\Archivos de programa\Aquila Technology\WakeOnLAN\WakeOnLanC.exe"
SET psexec="C:\Documents and Settings\OpenStack\Escritorio\soft\psexec"
SET adminUser=admin
SET adminPass=
SET openstackUser=openstack
SET openstackPass=
SET vmrun="C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"
SET controllerVM="D:\OpenStack\Controller\Controller.vmx"
SET networkVM="D:\OpenStack\Network\Network.vmx"
SET computeVM="D:\OpenStack\Compute1\Compute1.vmx"

```

Es tracta dels noms dels fitxers que contenen les dades per treballar, els usuaris, i els *paths* pels programes que ha d'utilitzar i de les màquines virtuals que ha d'arrencar.

A continuació s'arrenquen totes les estacions via *WakeOnLan* i aprofito per carregar credencials administratives per poder executar programes a les màquines remotes com a tal:

```

REM WAKE UP OF THE HOSTS
REM First we need admin credentials to use Pstools on remote hosts
REM Iterate through every host contained in hostsFile
FOR /F "usebackq tokens=1-3" %%A IN (%hostsFile%) DO (
    REM Gain credentials for rDesktop
    CMDKEY /generic:TERMSRV/%%A /user:%%A\adminUser% /pass:%adminPass%
    REM Gain credentials for transferring data
    CMDKEY /add:%%A /user:%%A\adminUser% /pass:%adminPass%
    REM Wake up hosts
    %wakeOnLan% -w -mac %%C
)

```

Després d'esperar un temps prudencial, s'arrenquen les màquines virtuals, començant pel *Controller*, seguidament pel *Network* i a continuació els 16 *Compute* restants, gràcies a les *VMware VIX (vmrun)*:

```

PING -n 120 127.0.0.1 >NUL:

REM WAKE UP OF THE VMS

REM Wake up Controller VM
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (
    %psexec% \\%%A -u %%A\openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %controllerVM%
)

PING -n 60 127.0.0.1 >NUL:

REM Wake up Network VM
FOR /F "usebackq tokens=1" %%A IN (%networkNode%) DO (
    %psexec% \\%%A -u %%A\openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %networkVM%
)

PING -n 60 127.0.0.1 >NUL:

REM Wake up Compute VMs
FOR /F "usebackq tokens=1" %%A IN (%computeNodes%) DO (
    %psexec% \\%%A -u %%A\openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %computeVM%
)
PING -n 30 127.0.0.1 >NUL:
)

```

Finalment, s'eliminen les credencials emmagatzemades:

```

REM When we're done, we don't need any more the credentials:
REM Remove the created credentials
FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (
    CMDKEY /delete:%%A
    REM Remove credentials for rDesktop
    CMDKEY /delete:TERMSRV/%%A
    REM Remove credentials for transferring data
    CMDKEY /delete:%%A
)

```

Pel que fa a l'aturada del *cloud*, el *batch file* comença exactament igual, carregant una sèrie de variables que necessita per treballar, i carregant les credencials necessàries:

```

SET hostsFile="hosts.txt"
SET computeNodes="computeNodes.txt"
SET networkNode="networkNode.txt"
SET controllerNode="controllerNode.txt"
SET psexec="C:\Documents and Settings\OpenStack\Escritorio\soft\psexec"
SET adminUser=admin
SET adminPass=
SET openstackUser=openstack
SET openstackPass=
SET vmrun="C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"
SET controllerVM="D:\OpenStack\Controller\Controller.vmx"
SET networkVM="D:\OpenStack\Network\Network.vmx"
SET computeVM="D:\OpenStack\Compute1\Compute1.vmx"
SET shutdownInstances="/home/santi/shutdownInstances.sh"

REM First we need admin credentials to use Pstools on remote hosts
REM Iterate through every host contained in hostsFile
FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (
    REM Gain credentials for rDesktop
    CMDKEY /generic:TERMSRV/%%A /user:%%A\adminUser% /pass:%adminPass%
    REM Gain credentials for transferring data
    CMDKEY /add:%%A /user:%%A\adminUser% /pass:%adminPass%
)

```

Tot seguit el que es fa és aturar les instàncies del *cloud*:

```

REM SHUTDOWN OF THE INSTANCES
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (
    %psexec% \\%%A -u %%A\openstackUser% -p %openstackPass% -h
    %vmrun% -T player -gu santi -gp 20openstack15 runProgramInGuest
    %controllerVM% %shutdownInstances%
)

```

Passat un temps prudencial, s'aturen les màquines virtuals en ordre invers al d'arrencada, és a dir, primer els de computació, després el de xarxa i finalment el controlador:



```

PING -n 30 127.0.0.1 >NUL:

REM SHUTDOWN OF THE VMS

REM Shutdown Compute VMs
FOR /F "usebackq tokens=1" %%A IN (%computeNodes%) DO (
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h
    %vmrun% -T player stop %computeVM% soft
)

PING -n 60 127.0.0.1 >NUL:

REM Shutdown Network VM
FOR /F "usebackq tokens=1" %%A IN (%networkNode%) DO (
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h
    %vmrun% -T player stop %networkVM% soft
)

PING -n 60 127.0.0.1 >NUL:

REM Shutdown Controller VM
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h
    %vmrun% -T player stop %controllerVM% soft
)

```

Finalment, i passat un temps prudencial també, s'aturen les estacions i s'eliminen les credencials:

```

PING -n 60 127.0.0.1 >NUL:

REM When we're done, we don't need any more the credentials:
REM Remove the created credentials
FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (
    REM Shutdown hosts
    shutdown /m \\%%A /s /t 0
    CMDKEY /delete:%%A
    REM Remove credentials for rDesktop
    CMDKEY /delete:TERMSRV/%%A
    REM Remove credentials for transferring data
    CMDKEY /delete:%%A
)

```

L'script que atura les màquines virtuals del *cloud* és quelcom tan senzill com:

```

source /home/santi/admin-openrc.sh
nova list --all_tenants | grep ACTIVE | awk '{print $2}' | xargs -I {} bash -c 'nova stop {}'

```

Finalment, vaig fer la versió tant del d'arrencada com del de parada que fa *log* de tot. El d'arrencada:

```

@echo off
echo. ----- Cloud started at %time% on %date% ----- >> start.log
start.bat >> start.log 2>&1

```

I el de parada:

```
@echo off
echo. ----- Cloud stoped at %time% on %date% ----- >> stop.log
stop.bat >> stop.log 2>&1
```

Als annexos es poden veure amb detall tots els *scripts* i fitxers involucrats.

### 9.3.2 Estacions

A les estacions de l'aula també s'han hagut de realitzar diverses tasques, entre elles:

1. Canvi del programa de congelació pel nou creat: descongelar, reiniciar, desinstal·lar el *deepfreeze* vell, reiniciar i instal·lar el nou.
2. Habilitació de l'escriptori remot.

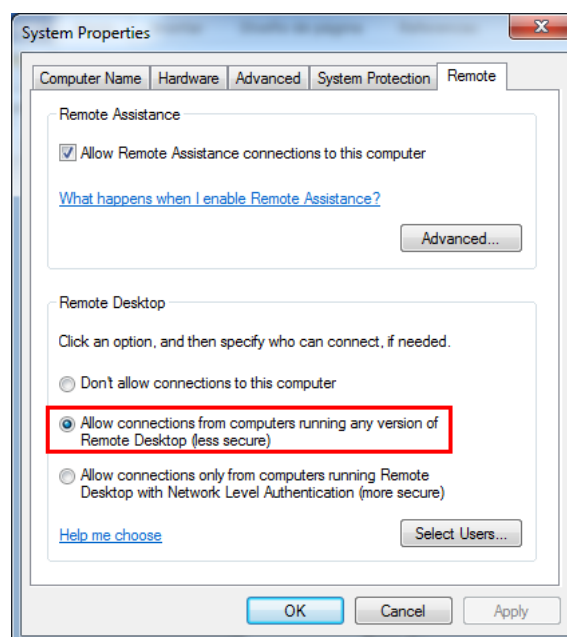


Figura 88: Accés remot Windows

3. Creació del nou usuari OpenStack.



Figura 89: Usuaris estacions

4. Creació d'una carpeta a la zona "descongelada" del disc amb permisos perquè només pugui accedir aquest usuari:

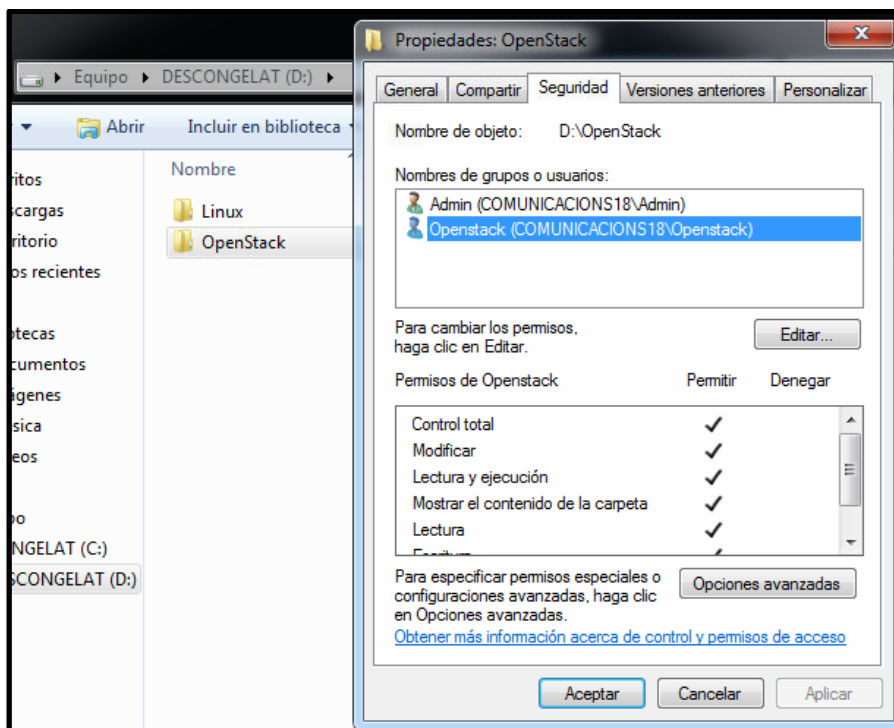


Figura 90: Permisos de la carpeta que conté la VM

5. Habilitar el *Wake on Lan* a les màquines.
6. Habilitar l'execució remota de comandes.
7. Instal·lar les VMware VIX. Per fer això només cal baixar el programa i seguir l'assistent d'instal·lació.
8. Habilitar el bit VT-x i configurar-lo també a l'*VMware Player*.
9. Crear un *script* d'adaptació de les màquines virtuals.

## Habilitació del Wake on Lan

Perquè una màquina es pugui arrencar a través de la xarxa, ha de:

- Disposar d'una targeta de xarxa que suporti el *Wake on Lan*, sent avui en dia, el cas més comú, i també el de les estacions de l'aula.
- Disposar d'una bios que permeti engegar la màquina responent a un esdeveniment d'un dispositiu PCI, PCI-E o bé directament *Wake on Lan*.

Els passos per configurar les bios de les estacions de l'aula són els següents:

1. Anar al mode *Advance*:



Figura 91: Pantalla inicial de la BIOS de les estacions

2. Seleccionar-lo a la finestra emergent:



Figura 92: Accés al mode Advanced

3. Un cop allà, seleccionem la pestanya *Advanced* i de la llista que apareix, la opció APM:

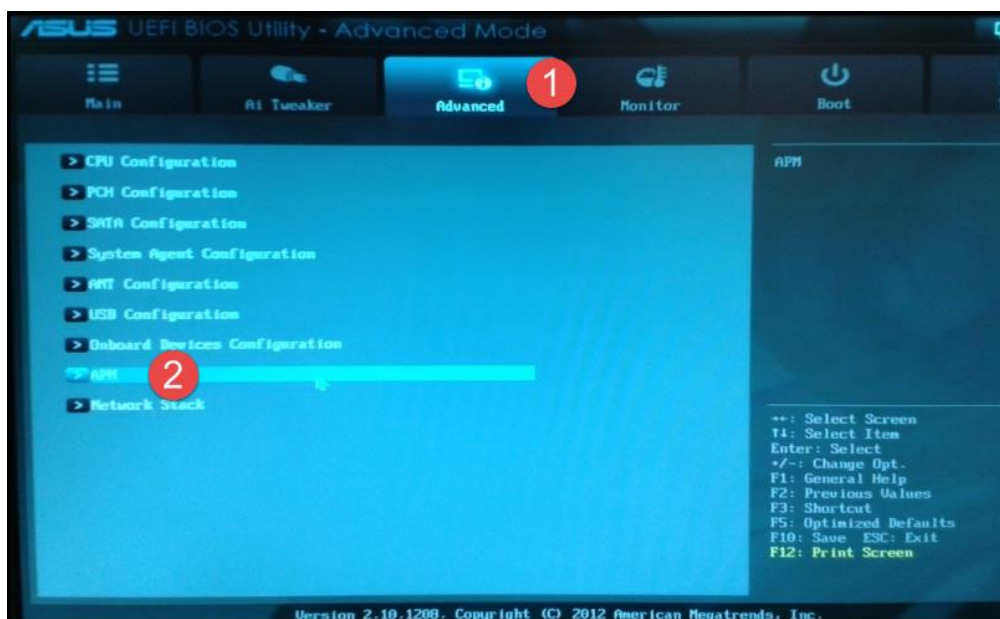


Figura 93: Accés a les propietats d'APM

4. Tot seguit, triem la opció “Power On By PCIE/PCI”, toquem al seu botó corresponent i triem la opció *Enabled* de la finestra emergent:

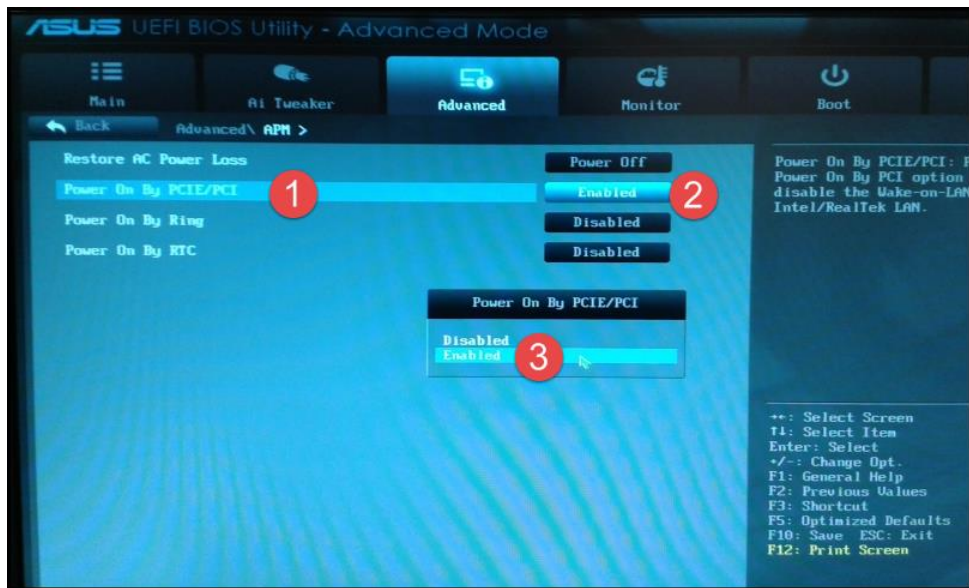


Figura 94: Activació del WakeOnLan

5. Per acabar, sortim guardant els canvis:

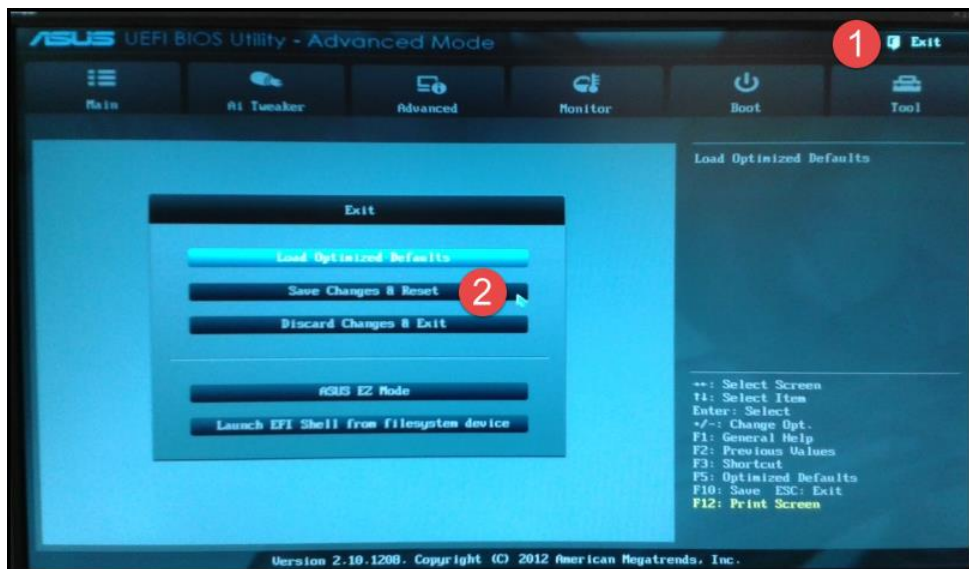


Figura 95: Guardar els canvis

Finalment, només cal configurar el sistema operatiu perquè permeti l'arrencada remota, cosa que es fa seguint els següents passos:

- En primer lloc cal anar a l'administrador de dispositius, cosa que s'aconsegueix executant (tecla Windows + r) la següent comanda:



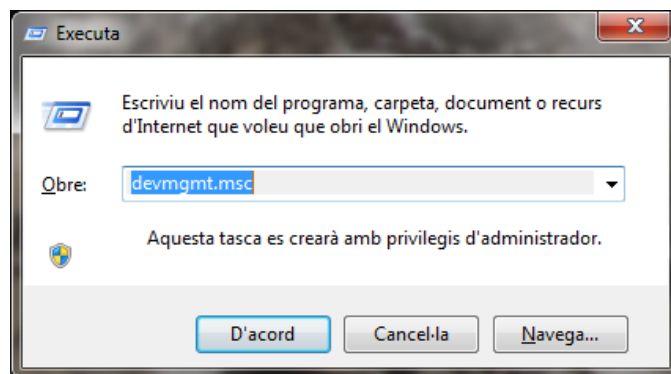


Figura 96: Accés a l'administrador de dispositius

- Un cop allà caldrà cercar la targeta de xarxa que haurà de rebre el *Magic Packet* que aixecarà la màquina i obrir-ne les propietats:

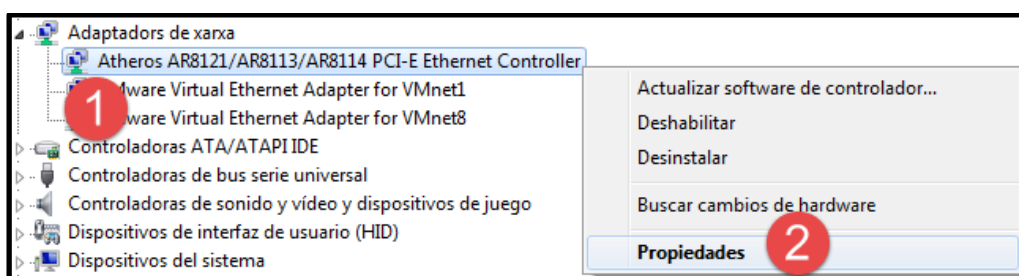


Figura 97: Accés a les propietats del dispositiu de xarxa

- Dins les propietats, s'ha de triar la pestanya "Avançats", de les propietats la de "Capacidades de reactivación", es tria el valor "**Magic Packet**" i finalment es prem l'OK:

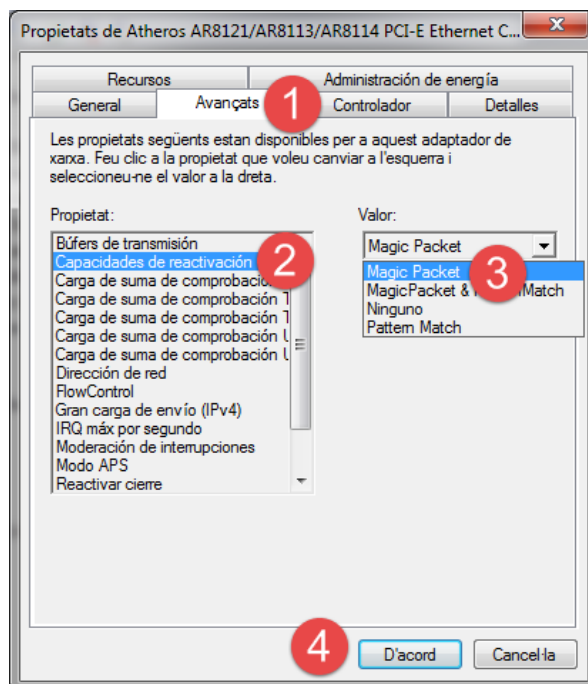


Figura 98: Propietats del dispositiu de xarxa

Seguint els passos descrits, ja tenim l'estació preparada, configuració que s'ha de replicar per les 18 màquines.

### Habilitació de l'execució remota de comandes (Pstools)

L'aula no estava preparada per poder fer servir les PsTools, l'eina per poder executar comandes remotament, i que em permetia administrar l'aula com un tot, sense haver d'anar màquina per màquina.

El procés per poder executar el component PsExec a les estacions ha estat el següent:

1. En primer lloc cal tenir habilitats els recursos compartits d'administració, que per defecte al Windows 7 estan deshabilitats (C\$, D\$, ..., admin\$), ja que aquesta eina fa servir el recurs "admin\$" per l'execució remota de les comandes. Per fer-ho, cal afegir una nova entrada al registre:

```
REG ADD \  
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System \  
/v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1
```

2. Obrir una nova entrada al firewall de Windows, ja que sinó tallarà totes les connexions referents als recursos compartits:

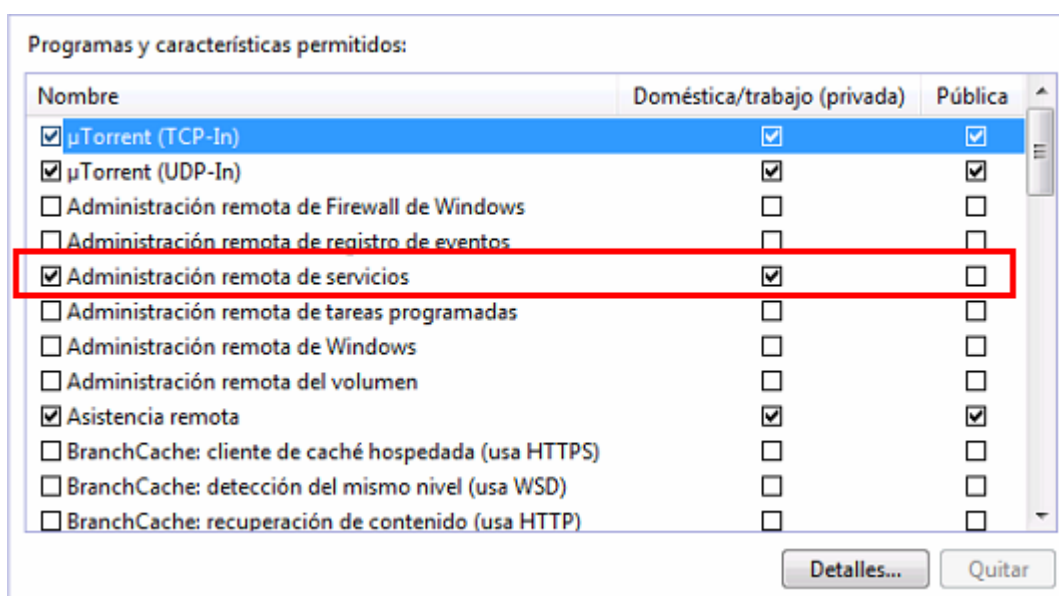


Figura 99: Excepció afegida al firewall

Un cop fets aquests petits canvis, ja es poden executar comandes de les estacions des d'un servidor remot.

### Habilitació de l'VT-x

Els requisits del *cloud* fan que l'habilitació d'aquest bit de la CPU sigui molt important, per dos motius:

- Els nodes de computació oferiran l'*hypervisor* KVM enlloc del QEMU. Aquest segon és una emulació del KVM.
- Si les màquines virtuals no disposen d'aquest bit habilitat a les seves CPUs virtuals, no es podrien crear màquines virtuals de més de 8GB de memòria.

El segon factor és important degut a que aquest *cloud* es vol utilitzar com a centre de càlcul, per tant, cal habilitar-lo. En primer lloc cal fer-ho a la bios de les estacions, els passos són els següents:

1. Anar al mode *Advance*:



Figura 100: Pantalla inicial de la BIOS

2. Seleccionar-lo a la finestra emergent:

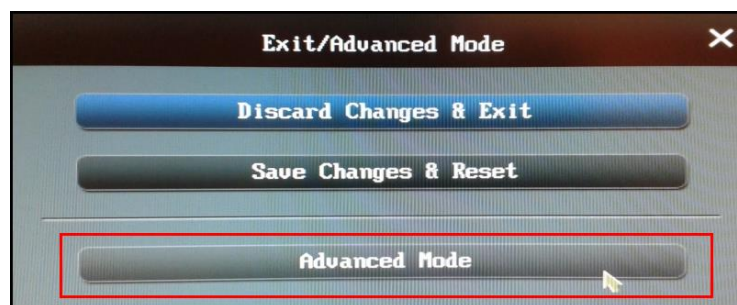


Figura 101: Accedir al mode advanced

3. Anar a *Advanced* i després a *CPU Configuration*:



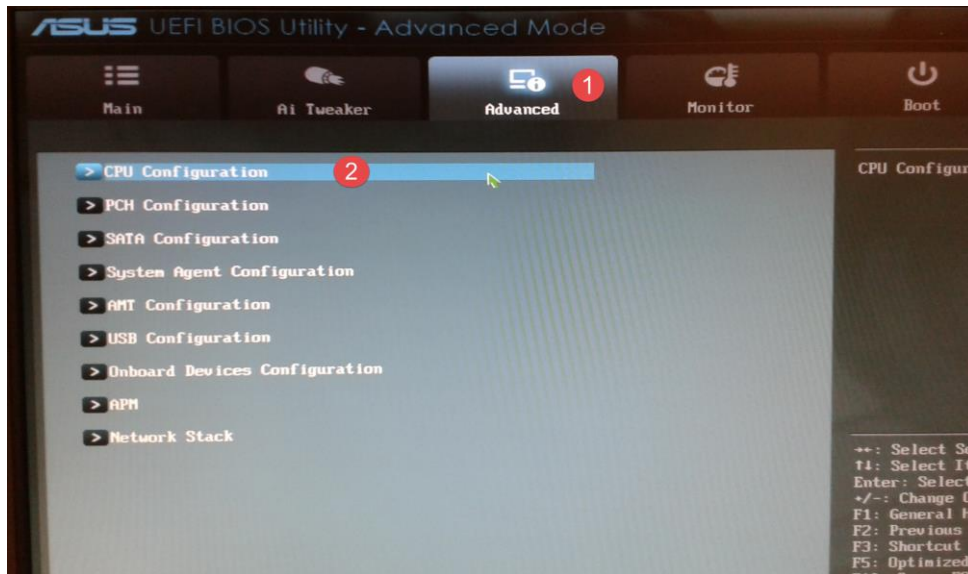


Figura 102: Accés a la configuració de la CPU

4. Habilitar *Intel Virtualization Technology*:

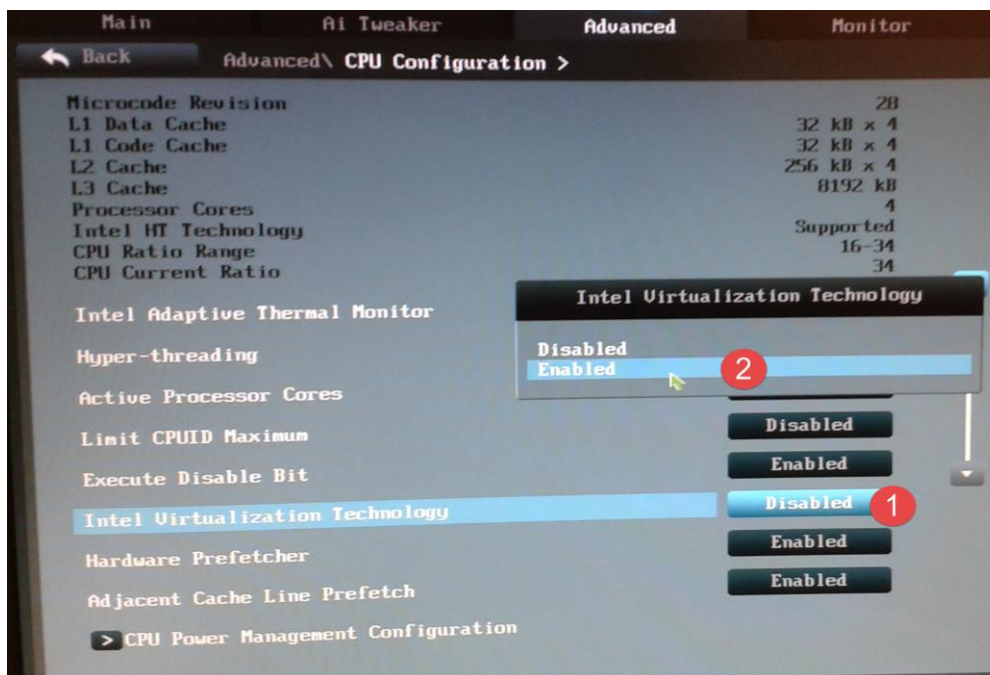


Figura 103: Activació del bit VT-x

5. Per acabar, sortim guardant els canvis:



Figura 104: Guardar els canvis

Seguidament, als VMware Player de les estacions cal habilitar-ho:

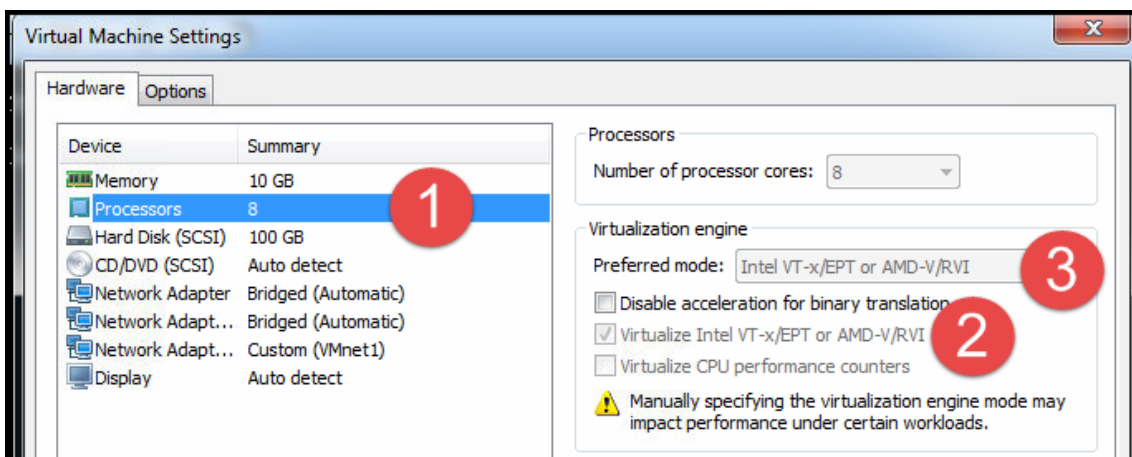


Figura 105: Activació de l'VT-x a l'VMware

I finalment modificar el valor del fitxer de configuració `/etc/nova/nova-compute.conf`, deixant-lo així:

```
[DEFAULT]
compute_driver=libvirt.LibvirtDriver
#[libvirt]
#virt_type = qemu
#virt_type = kvm
```

### Script d'adaptació de les màquines virtuals

Degut a la configuració de xarxa de l'aula, hi havia moltes configuracions que s'havien de variar dependent del node que es tractés. Els canvis a tenir en compte eren:

1. Calia que el fitxer **/etc/hosts** reflectís la topologia que es volia implementar a l'aula. Concretament calia posar la correspondència entre *hostnames* i IPs de la xarxa **Management** de cada node.
2. El fitxer **/etc/hostname** s'havia d'adaptar al node que s'estigués configurant. Més concretament, el que calia fer era posar el nom que li toqués. En aquest cas, no calia tocar ni el node **Controller** ni el node **Network**, ja que havien de tenir aquest nom per defecte. En canvi, els que s'havien de configurar eren els nodes **Compute**, pels quals calia anar assignant noms tals com **ComputeXX** on  $X \in \{01, 02, \dots, 16\}$ .
3. **NTP**. No calia fer canvis, ja que també hi havia redireccions als *servers* d'Ubuntu.
4. **MARIADB**. Calia editar el fitxer **/etc/mysql/my.cnf**, havent de configurar el paràmetre **bind-address** amb la IP del **Controller** a la xarxa *Management*, en el cas de l'aula és la **10.1.1.18**.
5. Servei **NOVA**. Al node **Controller**, calia editar el fitxer **/etc/nova/nova.conf**, a la secció [DEFAULT], el paràmetre **my\_ip**, que en el cas de l'aula havia de ser **10.1.1.18**.
6. Servei **NOVA**. Al node **Controller**, calia editar el fitxer **/etc/nova/nova.conf**, a la secció [DEFAULT], els paràmetres **vncserver\_listen** i **vncserver\_proxyclient\_address**, que en el cas de l'aula havia de ser **10.1.1.18**.
7. Servei **NOVA**. Al node **ComputeXX**, calia editar el fitxer **/etc/nova/nova.conf**, a la secció [DEFAULT], el paràmetre **my\_ip**, que en el cas de l'aula havia de ser **10.1.1.X**.
8. Servei **NOVA**. Al node **ComputeXX**, calia editar el fitxer **/etc/nova/nova.conf**, a la secció [DEFAULT], el paràmetre **vncserver\_proxyclient\_address**, que en el cas de l'aula havia de ser **10.1.1.16**.
9. Servei **NEUTRON**. Al node **Network**, calia anar al fitxer **/etc/neutron/plugins/ml2/ml2\_conf.ini** i configurar el paràmetre **local\_ip** de la secció [ovs] amb la IP **10.1.2.17**, que és la IP que li correspon al node **Network** a la xarxa **Tunnel** executant-se a l'ordinador **Comunicacions17**.
10. Servei **NEUTRON**. Al node **ComputeXX**, calia editar el fitxer **/etc/neutron/plugins/ml2/ml2\_conf.ini** a la secció [ovs], paràmetre **local\_ip**, on calia posar la ip del node **ComputeX** a la xarxa **Tunnel** (10.1.2.31 en el cas del **Compute01**).
11. Servei **CEILOMETER**. Al node **Controller** calia posar la IP correcte pel servidor de MongoDB. Concretament, al fitxer **/etc/mongodb.conf** calia canviar el paràmetre **bind\_ip** amb la IP que tocava pel **Controller** a la xarxa **Management**. En aquest cas, com que la màquina triada era la **Comunicacions18**, la IP era la **10.1.1.18**.

Doncs bé, totes aquestes tasques haguessin estat molt costoses en temps, així que vaig decidir realitzar un *script* (*setNodeParams.sh*) que donada una topologia de xarxa definida en un fitxer *hosts*, aquest s'encarregués de configurar el node desitjat a la màquina on s'estigués configurant.

El fitxer model de *hosts* per la topologia de xarxa actual (explicada a l'apartat 8.3, el disseny de l'escenari final) té aquest aspecte:

```
127.0.0.1 localhost

# OpenStack nodes
10.1.1.1 Compute01 # Comunicacions01 84.88.155.91
10.1.1.2 Compute02 # Comunicacions02 84.88.155.92
10.1.1.3 Compute03 # Comunicacions03 84.88.155.93
10.1.1.4 Compute04 # Comunicacions04 84.88.155.94
10.1.1.5 Compute05 # Comunicacions05 84.88.155.95
10.1.1.6 Compute06 # Comunicacions06 84.88.155.96
10.1.1.7 Compute07 # Comunicacions07 84.88.155.97
10.1.1.8 Compute08 # Comunicacions08 84.88.155.98
10.1.1.9 Compute09 # Comunicacions09 84.88.155.99
10.1.1.10 Compute10 # Comunicacions10 84.88.155.100
10.1.1.11 Compute11 # Comunicacions11 84.88.155.101
10.1.1.12 Compute12 # Comunicacions12 84.88.155.102
10.1.1.13 Compute13 # Comunicacions13 84.88.155.103
10.1.1.14 Compute14 # Comunicacions14 84.88.155.104
10.1.1.15 Compute15 # Comunicacions15 84.88.155.105
10.1.1.16 Compute16 # Comunicacions16 84.88.155.106
10.1.1.17 Network # Comunicacions17 84.88.155.107
10.1.1.18 Controller # Comunicacions18 84.88.155.108

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Figura 106: Contingut fitxer hosts

El codi de l'*script* es pot consultar amb detall a l'apartat d'annexos. Perquè aquest funcioni, només cal definir el fitxer de *hosts* amb el format de la imatge anterior, tenir l'*script* i el fitxer de *hosts* al mateix directori de la màquina a configurar i passar-li com a paràmetre el nom de la màquina real on s'està, en el format ComunicacionsXX.

## 10. Implantació i resultats

El punt de partida de la implantació a l'aula era el de tres màquines virtuals configurades de la següent manera:

- El node *Controller* es trobava a la màquina real Comunicacions18.
- El node *Network* es trobava a la màquina real Comunicacions17.
- El node *Compute* es trobava a la màquina real Comunicacions16.

Vaig copiar per cadascuna d'elles l'*script* de configuració *setNodeParams.sh* amb el fitxer *hosts* al directori */home/santi/nodeConfig*. Un cop copiat l'*script*, només calia executar-lo i parar la màquina virtual corresponent.

Un cop configurats adientment cada tipus de node, vaig fer còpies de seguretat d'aquests a diversos llocs. Fetes les còpies de seguretat, el pas que quedava fer era copiar el node *Compute* a la resta d'estacions, és a dir de la Comunicacions01 a la Comunicacions15.

Per fer les còpies de la màquina virtual a les estacions, ho vaig fer remotament, optimitzant les còpies de manera que les màquines que ja havien rebut la seva, passaven a servir a d'altres que encara no la tenien, fent còpies en paral·lel, per optimitzar el temps.

Un cop la resta d'estacions tenien el seu node de computació, només restava arrencar cadascuna, executar l'*script* de configuració i arrencar el *cloud* en l'ordre adient (*Controller*, *Network* i *Compute*). La primera arrencada la vaig fer manual per poder portar-ne el control. I el resultat va ser bo:

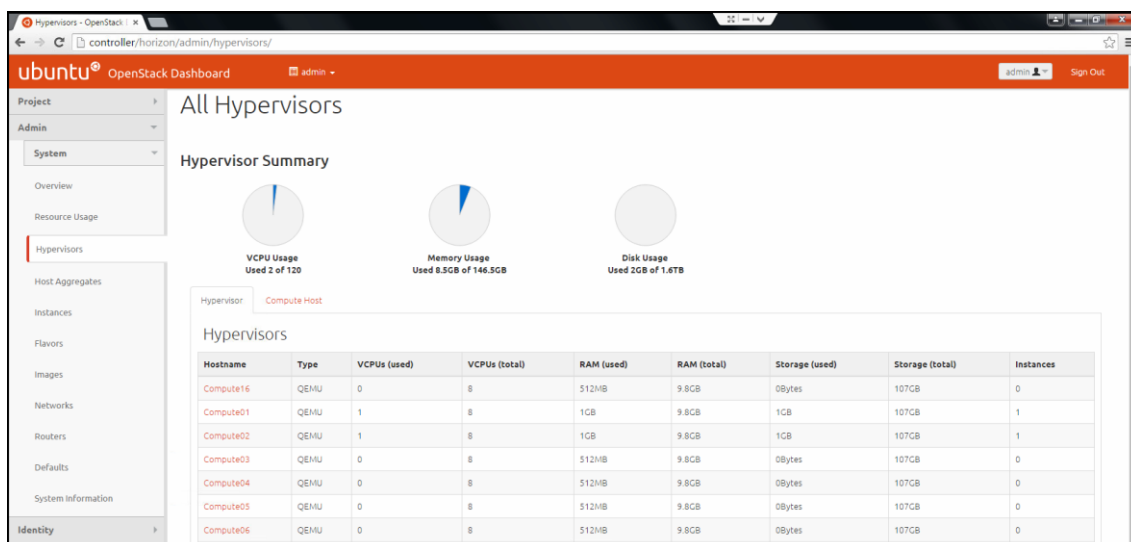


Figura 107: Vista de tots els recursos del cloud disponibles des de l'administrador

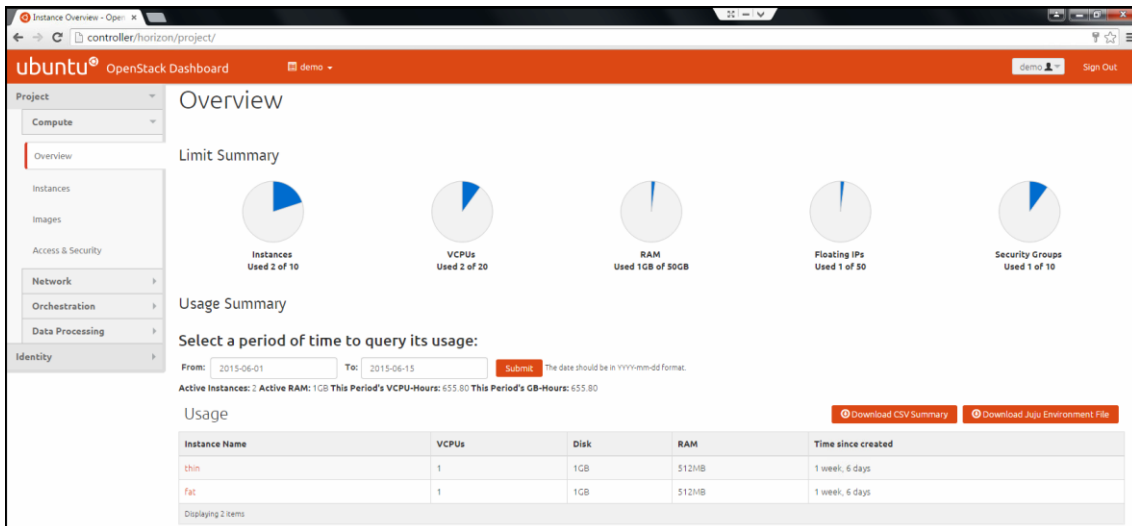


Figura 108: Vista de l'usuari

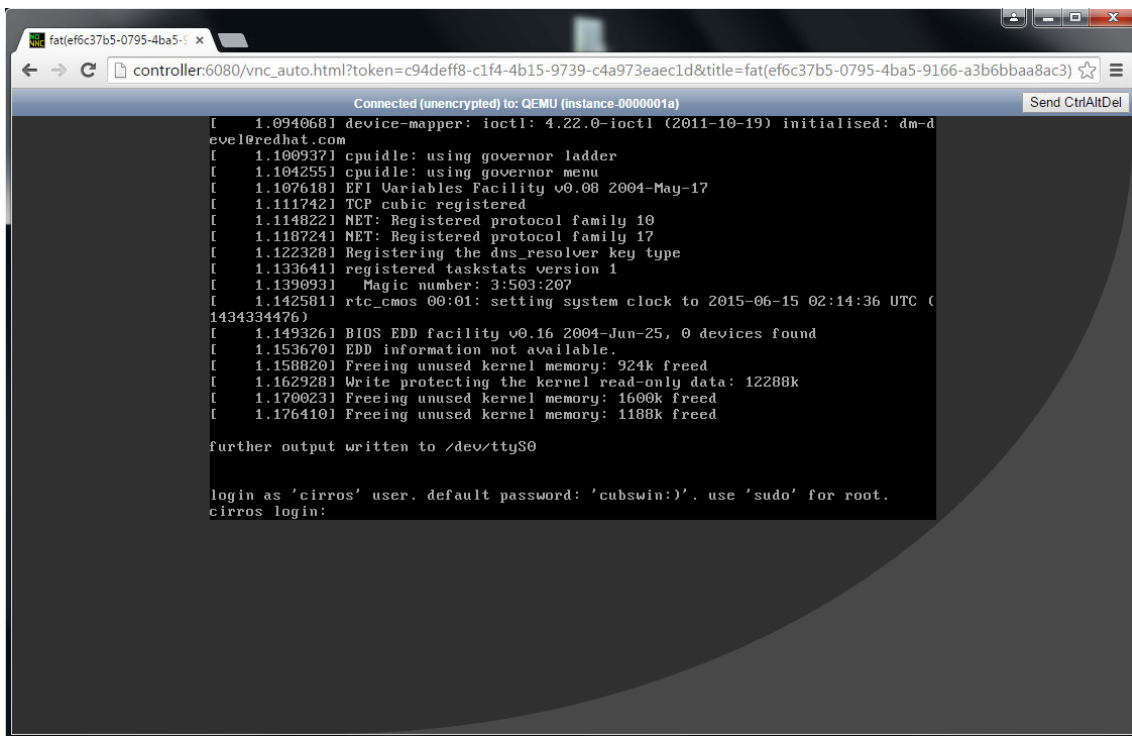


Figura 109: Vista de la consola de la seva instància

Finalment, només restava provar l'arrencada i la parada automàtica del *cloud*, simplement executant els *scripts* continguts al servidor *OpenStack2003*, i esperar els mateixos resultats, és a dir, poder accedir al *Dashboard* del *cloud*, poder administrar-lo i provar que un usuari pogués arrencar una instància.

Si comencem per l'arrencada, podem veure en el rellotge de la màquina de la següent imatge que es comença a les 5:25, amb totes les màquines parades. Es pot veure també a la barra de tasques com es va executant l'*script*.



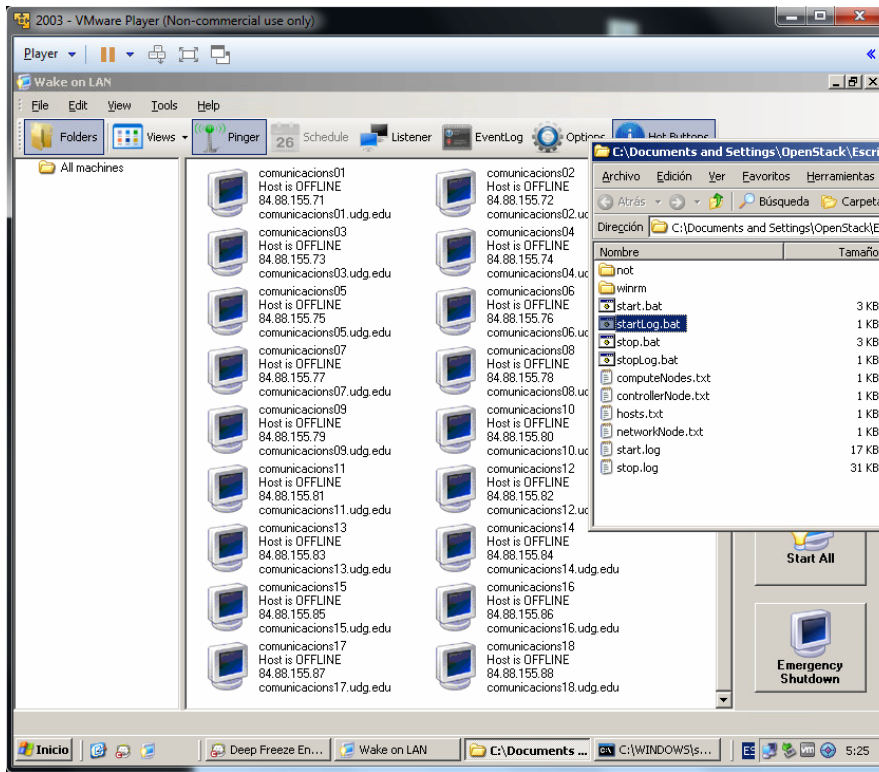


Figura 110: 1er pas executem l'script amb les màquines parades

Com podem comprovar a la següent imatge, després de 18 minuts, s'ha acabat d'executar l'*script* d'arrencada del *cloud*:

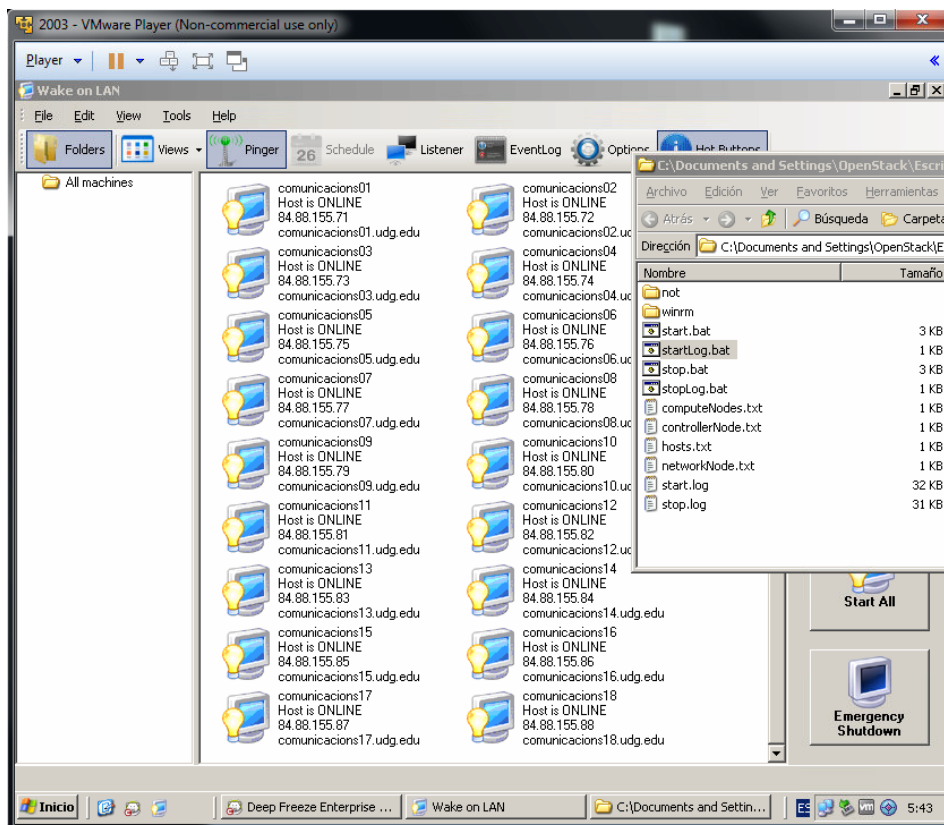


Figura 111: 2n pas, s'han encès totes les màquines (estacions i VMs)

I efectivament, l'usuari pot entrar al seu entorn d'administració:

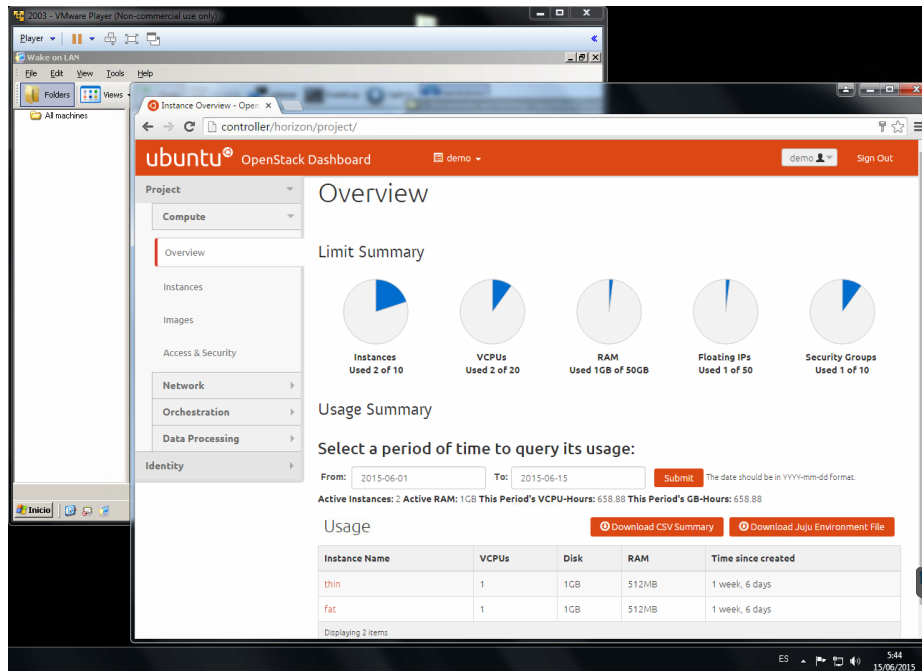


Figura 112: 3er pas, accés al Dashboard

Si comparem amb l'*stop* del *cloud*, aquest triga significativament menys. Comencem per executar l'*script* de parada, a les 5:18. Com es pot comprovar a la següent imatge, totes les màquines estan engegades i a la barra de tasques s'ha començat a executar l'*script* d'aturada.

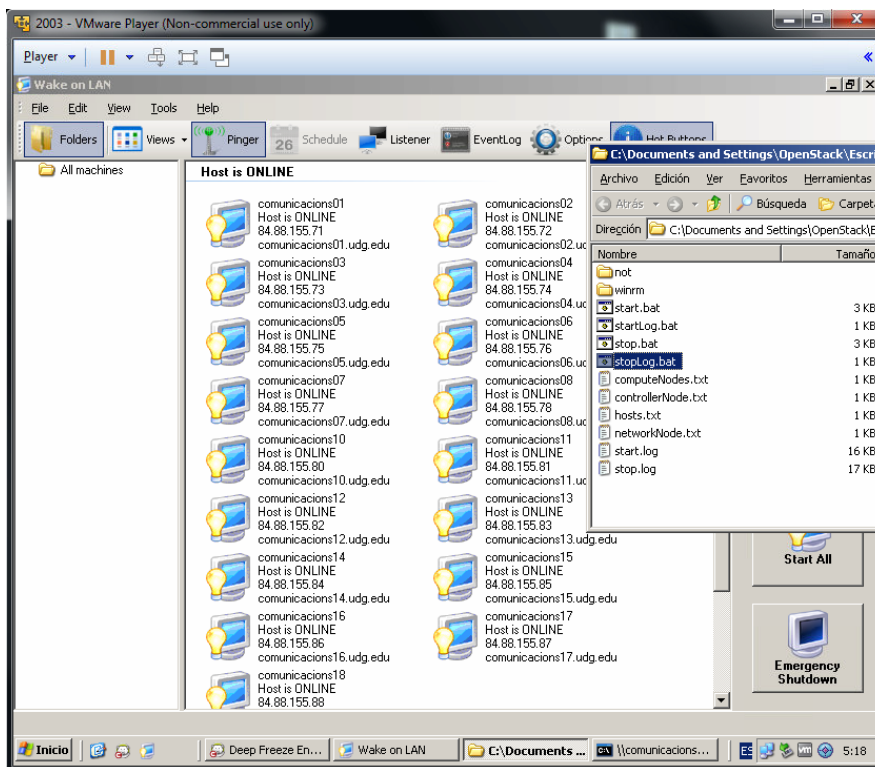


Figura 113: 1er pas, executar l'script amb totes les màquines engegades



Després de 5 minuts, l'*script* ha acabat i totes les màquines estan parades:

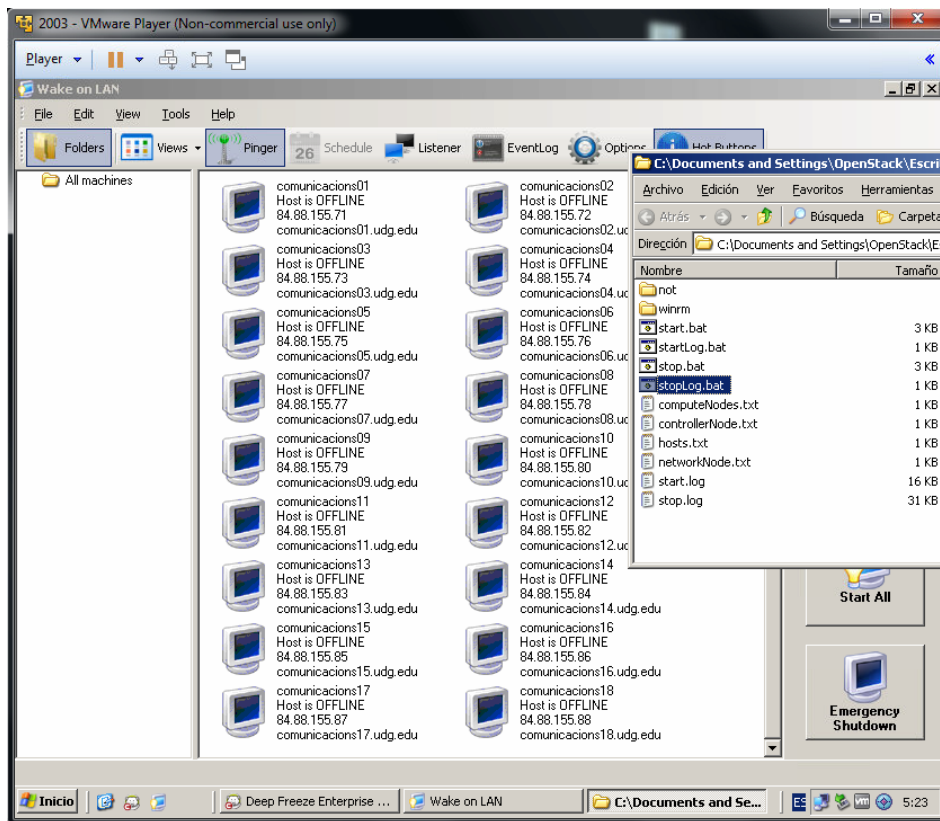


Figura 114: 2on pas, s'han aturat totes les màquines

I si intentem accedir al *Dashboard*, veiem que no tenim *cloud*:

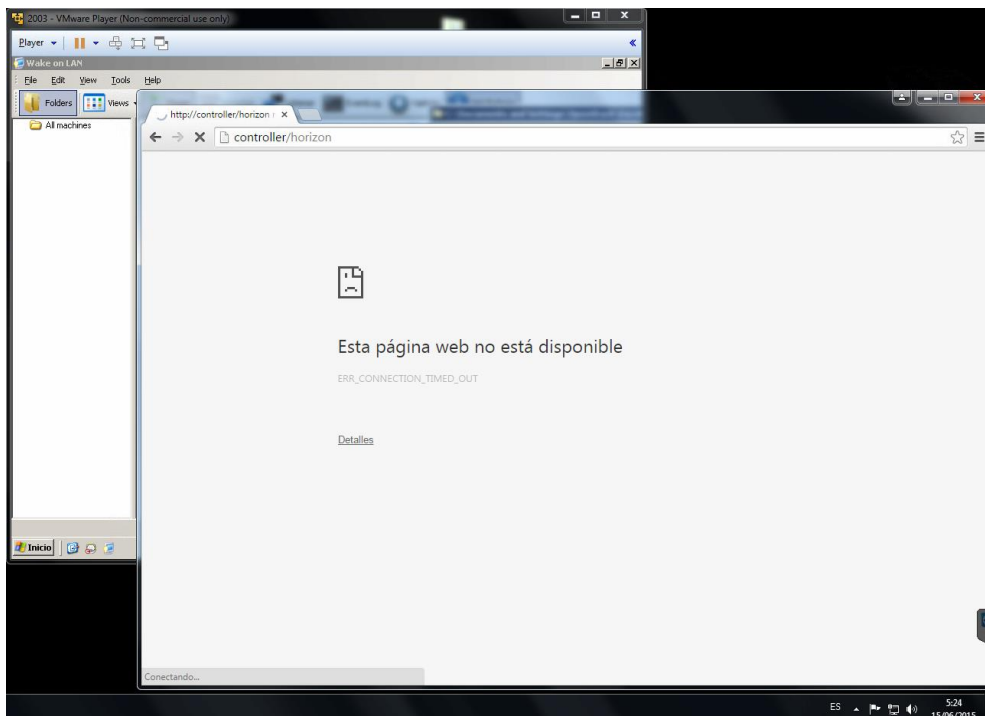


Figura 115: 3er pas, no es pot accedir al Dashboard

Aquestes tasques es poden programar fàcilment des del planificador de tasques del servidor; degut a que durant l'estiu no hi ha classe, deixarem *cloud* en mode d'arrencada "manual".

## 11. Conclusions

L'objectiu d'aquest projecte era reconvertir l'aula III-04i per poder desenvolupar les tasques de *cloud* i aula docent independentment, cosa que s'ha aconseguit gràcies a la virtualització, ja que de no haver estat així, segurament, el *cloud* hagués condicionat clarament els dos rols entre sí, provocant dependències entre ells.

Així, tenim que:

- Els nodes de l'OpenStack s'executen en un entorn aïllat (virtual) que no afecta al correcte funcionament dels sistemes operatius instal·lats a les estacions de l'aula.
- A més, gràcies a l'eina de congelació, tenim un punt més de seguretat vers modificacions inesperades del sistema operatiu, fet que ens dona la seguretat que l'*VMware Player* sempre funcionarà correctament.

També s'ha assolit la part d'habilitar el control total remot tant de les màquines reals com de les estacions:

- Per les estacions s'han habilitat eines con la consola del *Deepfreeze*, l'escriptori remot, les *PsTools* i nombrosos *scripts*.
- Les màquines virtuals es poden controlar remotament gràcies a l'ajuda de les *PsTools* i de les *VMware VIX*.
- Els recursos del *cloud* són administrables directament des de l'aplicació web.
- S'han programat *scripts* que permeten l'arrencada i la parada remotes de l'aula com a *cloud*.
- De la mateixa manera, s'han deixat les eines suficients com per poder realitzar un manteniment automatitzat i remot de l'aula.

Un dels altres punts a tenir en compte, és el *Dashboard*, l'aplicació web que permet controlar els recursos del *cloud* tan a nivell administratiu com d'usuari.

Finalment, el muntatge realitzat d'OpenStack, ha quedat encarat clarament com a centre de càlcul, elàstic gràcies als serveis de telemetria i per poder muntar-hi clústers virtuals, gràcies al servei *Sahara*.

Tot i que s'hagin assolit els objectius, el projecte es pot millorar en diversos aspectes que, per falta de temps, no s'han pogut tractar amb la profunditat requerida. A continuació, en el següent apartat de treball futur, es fan constar els punts més crítics a tenir en compte.

## 12. Treball futur

Existeixen diversos fronts que s'haurien d'abordar, per poder millorar diversos aspectes d'aquest *cloud*:

- La política que contrasenyes que vaig seguir va ser la de posar-les ben fàcils perquè d'una banda estava en un entorn de proves, i d'altra banda, qui hagi de venir després de mi no tingui dificultats d'accés. Això no vol dir que sigui una bona pràctica, ja que fa que l'entorn sigui insegur, així que, la primera cosa que caldria fer és canviar totes les contrasenyes del sistema per unes que compleixin els estàndards de seguretat.
- Moltes de les contrasenyes dels serveis de l'OpenStack es desen en text pla als fitxers de configuració. Caldria revisar com es podria evitar aquest forat de seguretat.
- Caldria que cada servei disposés d'un usuari per treballar amb el servei de missatgeria *RabbitMQ*.
- Les diferents xarxes que intervenen al sistema utilitzen el mateix medi físic, caldria separar-les en xarxes separades, i a poder ser, amb velocitats de com a mínim 1Gbps.
- La xarxa *External* del *cloud* es va quedar amb la configuració inicial. Caldria configurar-la per poder funcionar correctament amb l'aula.
- Fer que el procés d'arrencada del *cloud* sigui més ràpid.
- Les *PsTools* estan bé per entorns de proves com és el cas, però són una eina molt insegura. D'una banda vaig haver d'habilitar recursos compartits de les estacions (C\$, D\$, Admin\$) que per defecte al Windows 7 ja venen deshabilitats, però el que és més greu és que s'envien les contrasenyes en text pla.
- Com a alternativa, vaig començar a treballar amb les eines natives de Windows, el *Windows Management Framework Core*, amb la utilitat *WinRM*. Aquesta eina en sí no proporciona execució remota de programes executables, donat que arriba a executar un programa en remot, però passat un *timeout*, la comanda mor i també mata tots els processos que hagi engegat a la màquina remota.
- La següent alternativa que vaig provar va ser el *Powershell* del Windows, però per falta de temps no vaig arribar gaire lluny. Crec que seria un bon camí d'exploració.
- Cal idear alguna política de *backups*, ja que als requeriments del treball no es contempla, però crec que és quelcom necessari.
- S'ha de comprovar si el sistema es pot actualitzar o no (tan la plataforma d'administració de *cloud* com els sistemes operatius dels nodes).

- S'haurien d'afegir més imatges al repositori del *cloud*, ja que ara mateix només n'hi ha una, la del *Cirros*.
- Caldria estudiar si seria convenient instal·lar un servei SSH pels nodes de l'OpenStack, donat que ara mateix no el tenen.

## 13. Bibliografia

- [1] OpenStack Foundation, [OpenStack Installation Guide for Ubuntu 14.04](http://docs.openstack.org/juno/install-guide/install/apt/content/), 2014, <http://docs.openstack.org/juno/install-guide/install/apt/content/>
- [2] OpenStack Foundation, [OpenStack End User Guide](http://docs.openstack.org/user-guide/index.html), 2014, <http://docs.openstack.org/user-guide/index.html>
- [3] OpenStack Foundation, [OpenStack Admin User Guide](http://docs.openstack.org/user-guide-admin/index.html), 2014, <http://docs.openstack.org/user-guide-admin/index.html>
- [4] OpenStack Foundation, [OpenStack Operations Guide](http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html), 2014, [http://docs.openstack.org/openstack-ops/content/openstack-ops\\_preface.html](http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html)
- [5] OpenStack Foundation, [OpenStack Command-Line Interface Reference](http://docs.openstack.org/cli-reference/content/ch_preface.html), 2014, [http://docs.openstack.org/cli-reference/content/ch\\_preface.html](http://docs.openstack.org/cli-reference/content/ch_preface.html)
- [6] OpenStack Foundation, [OpenStack Training Guides](http://docs.openstack.org/icehouse/training-guides/content/), 2014, <http://docs.openstack.org/icehouse/training-guides/content/>
- [7] Kevin Jackson, [Installing RabbitMQ for OpenStack Cloud Computing Cookbook](http://openstackcookbook.com/), 2015, <http://openstackcookbook.com/>
- [8] OpenStack Foundation, [OpenStack Security Guide](http://docs.openstack.org/security-guide/content/ch_preface.html), 2014, [http://docs.openstack.org/security-guide/content/ch\\_preface.html](http://docs.openstack.org/security-guide/content/ch_preface.html)
- [9] OpenStack Foundation, [OpenStack command-line interface cheat sheet](http://docs.openstack.org/user-guide/cli_cheat_sheet.html), 2014, [http://docs.openstack.org/user-guide/cli\\_cheat\\_sheet.html](http://docs.openstack.org/user-guide/cli_cheat_sheet.html)
- [10] OpenStack Foundation, [OpenStack Storage](https://www.openstack.org/software/openstack-storage/), 2015, <https://www.openstack.org/software/openstack-storage/>
- [11] OpenStack Foundation, [Swift Architectural Overview](http://docs.openstack.org/developer/swift/overview_architecture.html), 2015, [http://docs.openstack.org/developer/swift/overview\\_architecture.html](http://docs.openstack.org/developer/swift/overview_architecture.html)
- [12] OpenStack Foundation, [OpenStack virtual Machine image guide](http://docs.openstack.org/image-guide/content/index.html), 2014, <http://docs.openstack.org/image-guide/content/index.html>
- [13] Ubuntu – Canonical, [Ubuntu OpenStack Reference Architecture](http://www.ubuntu.com/cloud/ubuntu-openstack), 2015, <http://www.ubuntu.com/cloud/ubuntu-openstack>
- [14] Canonical, [Juju Machine View: more control at your fingertips](https://insights.ubuntu.com/2014/09/26/juju-machine-view-more-control-at-your-fingertips/), 2014, <https://insights.ubuntu.com/2014/09/26/juju-machine-view-more-control-at-your-fingertips/>
- [15] Canonical, [Ubuntu OpenStack factsheet](https://insights.ubuntu.com/2013/08/29/ubuntu-openstack-factsheet/), 2013, <https://insights.ubuntu.com/2013/08/29/ubuntu-openstack-factsheet/>
- [16] pfSense, [pfSense 2 on VMware ESXi 5](https://doc.pfsense.org/index.php/PfSense_2_on_VMware_ESXi_5), 2015, [https://doc.pfsense.org/index.php/PfSense\\_2\\_on\\_VMware\\_ESXi\\_5](https://doc.pfsense.org/index.php/PfSense_2_on_VMware_ESXi_5)
- [17] Navesta (Ubuntu User), [Openstack Juno Sahara Data Processing on 14.04](http://askubuntu.com/questions/555093/openstack-juno-sahara-data-processing-on-14-04), 2014, <http://askubuntu.com/questions/555093/openstack-juno-sahara-data-processing-on-14-04>
- [18] OpenStack Foundation, [Sahara Installation Guide](http://docs.openstack.org/developer/sahara/userdoc/installation.guide.html), 2014, <http://docs.openstack.org/developer/sahara/userdoc/installation.guide.html>
- [19] OpenStack Foundation, [Sahara Configuration Guide](http://docs.openstack.org/developer/sahara/userdoc/configuration.guide.html), 2015, <http://docs.openstack.org/developer/sahara/userdoc/configuration.guide.html>
- [20] Red Hat, Inc., [RDO Quickstart](https://www.rdoproject.org/Quickstart), 2014, <https://www.rdoproject.org/Quickstart>
- [21] Mirantis, Inc., [Quick start guide](https://software.mirantis.com/quick-start/), 2015, <https://software.mirantis.com/quick-start/>
- [22] Tim Ferrill, [Powershell Basics: Managing remote Servers & Sessions](http://www.tomsitpro.com/articles/powershell-remote-management,2-781.html), 2014, <http://www.tomsitpro.com/articles/powershell-remote-management,2-781.html>

- [23] Chris Hoffman, [How to Run PowerShell Commands on Remote Computers](http://www.howtogeek.com/117192/how-to-run-powershell-commands-on-remote-computers/), 2012, <http://www.howtogeek.com/117192/how-to-run-powershell-commands-on-remote-computers/>
- [24] JPBlanc, [Launching background tasks in a remote session that don't get killed when the session is removed](http://stackoverflow.com/questions/8675847/launching-background-tasks-in-a-remote-session-that-dont-get-killed-when-the-se/8704945#8704945), 2012, <http://stackoverflow.com/questions/8675847/launching-background-tasks-in-a-remote-session-that-dont-get-killed-when-the-se/8704945#8704945>
- [25] Christopher Neylan, [Invoke background command on remote Machine](http://stackoverflow.com/questions/11728192/invoke-background-command-on-remote-machine), 2012, <http://stackoverflow.com/questions/11728192/invoke-background-command-on-remote-machine>
- [26] Pavel Chuchuva, [How to enable execution of Powershell scripts?](http://superuser.com/questions/106360/how-to-enable-execution-of-powershell-scripts), 2014, <http://superuser.com/questions/106360/how-to-enable-execution-of-powershell-scripts>
- [27] Mike Plkington, [The Power of Powershell Remoting](http://digital-forensics.sans.org/blog/2013/09/03/the-power-of-powershell-remoting), 2013, <http://digital-forensics.sans.org/blog/2013/09/03/the-power-of-powershell-remoting>
- [28] Po\$H Pete, [Passing username and passwords to remote desktop \(RDP\) automatically](http://www.poshpete.com/powershell/passing-username-and-passwords-to-remote-desktop-rdp-automatically), 2011, <http://www.poshpete.com/powershell/passing-username-and-passwords-to-remote-desktop-rdp-automatically>
- [29] SS64.com, [CMD, Bash & PowerShell](http://ss64.com/), 2015, <http://ss64.com/>
- [30] Ben Griswold, [How do you run CMD.exe under the Local System Account?](http://stackoverflow.com/questions/77528/how-do-you-run-cmd-exe-under-the-local-system-account), 2008, <http://stackoverflow.com/questions/77528/how-do-you-run-cmd-exe-under-the-local-system-account>
- [31] Evan Anderson, [Batch file to MASS ping group of computers on network by name, check reply, and resolve hostname](http://serverfault.com/questions/85372/batch-file-to-mass-ping-group-of-computers-on-network-by-name-check-reply-and-resolve-hostname), 2009, <http://serverfault.com/questions/85372/batch-file-to-mass-ping-group-of-computers-on-network-by-name-check-reply-and-resolve-hostname>
- [32] Paxdiablo, [How to check if ping responded or not in a batch file](http://stackoverflow.com/questions/3050898/how-to-check-if-ping-responded-or-not-in-a-batch-file), 2010, <http://stackoverflow.com/questions/3050898/how-to-check-if-ping-responded-or-not-in-a-batch-file>
- [33] Dbenham, [Batch file call sub-batch file to pass n paramètres and return without use of file](http://stackoverflow.com/questions/11481150/batch-file-call-sub-batch-file-to-pass-n-parametres-and-return-without-use-of-file), 2012, [http://stackoverflow.com/questions/11481150/batch-file-call-sub-batch-file-to-pass-n-parametres-and-return-without-use-of-f](http://stackoverflow.com/questions/11481150/batch-file-call-sub-batch-file-to-pass-n-parametres-and-return-without-use-of-file)
- [34] Windows command Line, [Add users, groups, xcopy](http://www.windows-commandline.com/add-user-from-command-line/), 2009, <http://www.windows-commandline.com/add-user-from-command-line/>
- [35] Computerboom, [Enable WOL wake on lan for Asus P5Q motherboard](http://computerboom.blogspot.com.es/2008/12/howto-enable-wol-wake-on-lan-for-asus.html), 2008, <http://computerboom.blogspot.com.es/2008/12/howto-enable-wol-wake-on-lan-for-asus.html>
- [36] Aquila Technology, [WakeOnLan Help](http://wol.aquilatech.com/help/command_line/default.html), 2015, [http://wol.aquilatech.com/help/command\\_line/default.html](http://wol.aquilatech.com/help/command_line/default.html)
- [37] Oliver Baty, [Using vmrun to control virtual Machines in VMware Workstation](https://ardamis.com/2012/03/07/using-vmrun-to-control-virtual-machines-in-vmware-workstation/), 2012, <https://ardamis.com/2012/03/07/using-vmrun-to-control-virtual-machines-in-vmware-workstation/>
- [38] VMware, inc., [Using vmrun to Control Virtual Machines](http://www.vmware.com/products/beta/ws/vmrunCommand.pdf), 2008, <http://www.vmware.com/products/beta/ws/vmrunCommand.pdf>
- [39] Bjohns36, [VMrun exit codes](https://communities.vmware.com/message/1635921), 2010, <https://communities.vmware.com/message/1635921>

- [40] Bunyk, [How to allow access to winrs for non-admin user?](http://serverfault.com/questions/590515/how-to-allow-access-to-winrs-for-non-admin-user), 2014, <http://serverfault.com/questions/590515/how-to-allow-access-to-winrs-for-non-admin-user>
- [41] JonJor, [WinRM \(Windows Remote management\) Troubleshooting](http://blogs.technet.com/b/jonjor/archive/2009/01/09/winrm-windows-remote-management-troubleshooting.aspx), 2009, <http://blogs.technet.com/b/jonjor/archive/2009/01/09/winrm-windows-remote-management-troubleshooting.aspx>
- [42] HopelessN00b, [How can I connect to a Windows server using a CLI?](http://serverfault.com/questions/429426/how-can-i-connect-to-a-windows-server-using-a-command-line-interface-cli), 2014, <http://serverfault.com/questions/429426/how-can-i-connect-to-a-windows-server-using-a-command-line-interface-cli>
- [43] Microsoft, inc., [Installation and Configuration for Windows Remote Management](https://msdn.microsoft.com/en-us/library/aa384372(v=vs.85).aspx), 2015, [https://msdn.microsoft.com/en-us/library/aa384372\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa384372(v=vs.85).aspx)
- [44] Kayasax, [How to add more than one Machine to the trusted hosts list using winrm](http://stackoverflow.com/questions/21548566/how-to-add-more-than-one-machine-to-the-trusted-hosts-list-using-winrm), 2014, <http://stackoverflow.com/questions/21548566/how-to-add-more-than-one-machine-to-the-trusted-hosts-list-using-winrm>
- [45] Mike Pilkington, [Protecting admin passwords during remote response and forensics](http://digital-forensics.sans.org/blog/2010/06/01/protecting-admin-passwords-remote-response-forensics/), 2010, <http://digital-forensics.sans.org/blog/2010/06/01/protecting-admin-passwords-remote-response-forensics/>
- [46] OpenStack Foundation, [User stories](http://digital-forensics.sans.org/blog/2010/06/01/protecting-admin-passwords-remote-response-forensics/), 2015, <http://digital-forensics.sans.org/blog/2010/06/01/protecting-admin-passwords-remote-response-forensics/>
- [47] OpenNebula.org, [VirtualBox Sandbox](http://openebula.org/tryout/sandboxvirtualbox/), 2015, <http://openebula.org/tryout/sandboxvirtualbox/>
- [48] Eucalyptus Systems, [Eucalyptus 3.4.2 User Guide](https://www.eucalyptus.com/docs/eucalyptus/3.4/user-guide-3.4.2.pdf), 2014, <https://www.eucalyptus.com/docs/eucalyptus/3.4/user-guide-3.4.2.pdf>
- [49] OpenNebula.org, [About the OpenNebula Technology](http://openebula.org/about/technology/), 2015, <http://openebula.org/about/technology/>
- [50] OpenNebula.org, [Planning the Installation](http://docs.openebula.org/4.12/design_and_installation/building_your_cloud/plan.html), 2015, [http://docs.openebula.org/4.12/design\\_and\\_installation/building\\_your\\_cloud/plan.html](http://docs.openebula.org/4.12/design_and_installation/building_your_cloud/plan.html)
- [51] Revista Cloud Computing, [Comparativa de las plataformas cloud abiertas](http://www.revistacloudcomputing.com/2013/01/comparativa-de-las-plataformas-cloud-abiertas-openstack-openebula-eucalyptus-y-cloudstack/), 2013, <http://www.revistacloudcomputing.com/2013/01/comparativa-de-las-plataformas-cloud-abiertas-openstack-openebula-eucalyptus-y-cloudstack/>



## 14. Annexos

### 14.1 Arrencada / Parada del Cloud

#### Fitxer computeNodes.txt

```
comunicacions01.udg.edu
comunicacions02.udg.edu
comunicacions03.udg.edu
comunicacions04.udg.edu
comunicacions05.udg.edu
comunicacions06.udg.edu
comunicacions07.udg.edu
comunicacions08.udg.edu
comunicacions09.udg.edu
comunicacions10.udg.edu
comunicacions11.udg.edu
comunicacions12.udg.edu
comunicacions13.udg.edu
comunicacions14.udg.edu
comunicacions15.udg.edu
comunicacions16.udg.edu
```

#### Fitxer controllerNode.txt

```
comunicacions18.udg.edu
```

#### Fitxer hosts.txt

```
comunicacions01.udg.edu 84.88.155.71 30:85:A9:97:C9:6D
comunicacions02.udg.edu 84.88.155.72 30:85:A9:97:C9:6F
comunicacions03.udg.edu 84.88.155.73 30:85:A9:97:C9:72
comunicacions04.udg.edu 84.88.155.74 30:85:A9:96:B4:76
comunicacions05.udg.edu 84.88.155.75 30:85:A9:97:C6:85
comunicacions06.udg.edu 84.88.155.76 30:85:A9:96:B5:B2
comunicacions07.udg.edu 84.88.155.77 30:85:A9:96:B5:69
comunicacions08.udg.edu 84.88.155.78 30:85:A9:96:B7:11
comunicacions09.udg.edu 84.88.155.79 30:85:A9:96:B5:5B
comunicacions10.udg.edu 84.88.155.80 30:85:A9:97:C9:8A
comunicacions11.udg.edu 84.88.155.81 30:85:A9:97:C9:5F
comunicacions12.udg.edu 84.88.155.82 30:85:A9:97:C8:EC
comunicacions13.udg.edu 84.88.155.83 30:85:A9:97:C9:18
comunicacions14.udg.edu 84.88.155.84 30:85:A9:96:B5:5A
comunicacions15.udg.edu 84.88.155.85 30:85:A9:96:C1:62
comunicacions16.udg.edu 84.88.155.86 30:85:A9:96:B4:C1
comunicacions17.udg.edu 84.88.155.87 30:85:A9:96:B5:6F
comunicacions18.udg.edu 84.88.155.88 30:85:A9:96:B5:52
```

#### Fitxer networkNode.txt

```
comunicacions17.udg.edu
```

#### Fitxer start.bat

```
@ECHO off
```

```
REM @SETLOCAL EnableDelayedExpansion EnableExtensions
```

```
REM
```

```
REM VARIABLES
```

```
REM -----
```

```

REM

SET hostsFile="hosts.txt"
SET computeNodes="computeNodes.txt"
SET networkNode="networkNode.txt"
SET controllerNode="controllerNode.txt"
SET wakeOnLan="C:\Archivos de programa\Aquila
Technology\WakeOnLAN\WakeOnLanC.exe"
SET psexec="C:\Documents and
Settings\OpenStack\Escritorio\soft\psexec"
SET adminUser=admin
SET adminPass=
SET openstackUser=openstack
SET openstackPass=
SET vmrun="C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"
SET controllerVM="D:\OpenStack\Controller\Controller.vmx"
SET networkVM="D:\OpenStack\Network\Network.vmx"
SET computeVM="D:\OpenStack\Compute1\Compute1.vmx"

REM WAKE UP OF THE HOSTS
REM First we need admin credentials to use Pstools on remote hosts
REM Iterate through every host contained in hostsFile
FOR /F "usebackq tokens=1-3" %%A IN (%hostsFile%) DO (
    REM Gain credentials for rDesktop
    CMDKEY /generic:TERMSRV/%%A /user:%%A\%adminUser%
/pass:%adminPass%
    REM Gain credentials for transferring data
    CMDKEY /add:%%A /user:%%A\%adminUser% /pass:%adminPass%
    REM Wake up hosts
    %wakeOnLan% -w -mac %%C
)

PING -n 120 127.0.0.1 >NUL:

REM WAKE UP OF THE VMS

REM Wake up Controller VM
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (
    %psexec% \\\%%A -u %%A\%openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %controllerVM%
)

PING -n 60 127.0.0.1 >NUL:

REM Wake up Network VM
FOR /F "usebackq tokens=1" %%A IN (%networkNode%) DO (
    %psexec% \\\%%A -u %%A\%openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %networkVM%
)

PING -n 60 127.0.0.1 >NUL:

REM Wake up Compute VMs
FOR /F "usebackq tokens=1" %%A IN (%computeNodes%) DO (
    %psexec% \\\%%A -u %%A\%openstackUser% -p %openstackPass% -h -i 0
    %vmrun% -T player start %computeVM%
    PING -n 30 127.0.0.1 >NUL:
)

REM When we're done, we don't need any more the credentials:
REM Remove the created credentials

```

```

FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (
    CMDKEY /delete:%%A
    REM Remove credentials for rDesktop
    CMDKEY /delete:TERMSRV/%%A
    REM Remove credentials for transferring data
    CMDKEY /delete:%%A
)

```

## Fitxer startLog.bat

```

@echo off
echo. ----- Cloud started at %time% on %date% ----- >>
start.log
start.bat >> start.log 2>&1

```

## Fitxer stop.bat

```

@ECHO off

REM @SETLOCAL EnableDelayedExpansion EnableExtensions

REM
REM VARIABLES
REM -----
REM

SET hostsFile="hosts.txt"
SET computeNodes="computeNodes.txt"
SET networkNode="networkNode.txt"
SET controllerNode="controllerNode.txt"
SET psexec="C:\Documents and
Settings\OpenStack\Escritorio\soft\psexec"
SET adminUser=admin
SET adminPass=
SET openstackUser=openstack
SET openstackPass=
SET vmrun="C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"
SET controllerVM="D:\OpenStack\Controller\Controller.vmx"
SET networkVM="D:\OpenStack\Network\Network.vmx"
SET computeVM="D:\OpenStack\Computel\Computel.vmx"
SET shutdownInstances="/home/santi/shutdownInstances.sh"

REM First we need admin credentials to use Pstools on remote hosts
REM Iterate through every host contained in hostsFile
FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (
    REM Gain credentials for rDesktop
    CMDKEY /generic:TERMSRV/%%A /user:%%A\%adminUser%
/pass:%adminPass%
    REM Gain credentials for transferring data
    CMDKEY /add:%%A /user:%%A\%adminUser% /pass:%adminPass%
)

REM SHUTDOWN OF THE INSTANCES
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h
%vmrun% -T player -gu santi -gp 20openstack15 runProgramInGuest
%controllerVM% %shutdownInstances%
)

PING -n 30 127.0.0.1 >NUL:

```

```
REM SHUTDOWN OF THE VMS
```

```
REM Shutdown Compute VMs
```

```
FOR /F "usebackq tokens=1" %%A IN (%computeNodes%) DO (  
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h  
    %vmlrun% -T player stop %computeVM% soft  
)
```

```
PING -n 60 127.0.0.1 >NUL:
```

```
REM Shutdown Network VM
```

```
FOR /F "usebackq tokens=1" %%A IN (%networkNode%) DO (  
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h  
    %vmlrun% -T player stop %networkVM% soft  
)
```

```
PING -n 60 127.0.0.1 >NUL:
```

```
REM Shutdown Controller VM
```

```
FOR /F "usebackq tokens=1" %%A IN (%controllerNode%) DO (  
    %psexec% \\%%A -u %%A\%openstackUser% -p %openstackPass% -h  
    %vmlrun% -T player stop %controllerVM% soft  
)
```

```
PING -n 60 127.0.0.1 >NUL:
```

```
REM When we're done, we don't need any more the credentials:
```

```
REM Remove the created credentials
```

```
FOR /F "usebackq tokens=1" %%A IN (%hostsFile%) DO (  
    REM Shutdown hosts  
    shutdown /m \\%%A /s /t 0  
    CMDKEY /delete:%%A  
    REM Remove credentials for rDesktop  
    CMDKEY /delete:TERMSRV/%%A  
    REM Remove credentials for transferring data  
    CMDKEY /delete:%%A  
)
```

## Fitxer stopLog.bat

```
@echo off  
echo. ----- Cloud stoped at %time% on %date% ----- >>  
stop.log  
stop.bat >> stop.log 2>&1
```

## 14.2 Fitxers d'*script* d'adaptació dels nodes

### Fitxer hosts

```
127.0.0.1 localhost
```

```
# OpenStack nodes
```

```
10.1.1.1 Compute01 # Comunicacions01 84.88.155.91  
10.1.1.2 Compute02 # Comunicacions02 84.88.155.92  
10.1.1.3 Compute03 # Comunicacions03 84.88.155.93  
10.1.1.4 Compute04 # Comunicacions04 84.88.155.94  
10.1.1.5 Compute05 # Comunicacions05 84.88.155.95  
10.1.1.6 Compute06 # Comunicacions06 84.88.155.96
```

```

10.1.1.7    Compute07    # Comunicacions07 84.88.155.97
10.1.1.8    Compute08    # Comunicacions08 84.88.155.98
10.1.1.9    Compute09    # Comunicacions09 84.88.155.99
10.1.1.10   Compute10    # Comunicacions10 84.88.155.100
10.1.1.11   Compute11    # Comunicacions11 84.88.155.101
10.1.1.12   Compute12    # Comunicacions12 84.88.155.102
10.1.1.13   Compute13    # Comunicacions13 84.88.155.103
10.1.1.14   Compute14    # Comunicacions14 84.88.155.104
10.1.1.15   Compute15    # Comunicacions15 84.88.155.105
10.1.1.16   Compute16    # Comunicacions16 84.88.155.106
10.1.1.17   Network      # Comunicacions17 84.88.155.107
10.1.1.18   Controller  # Comunicacions18 84.88.155.108

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

```

## Fitxer setNodeParams.sh

```

#!/bin/bash

#
# This script makes all needed changes to any node
# configuration files to work into a real machine in the
# computer classroom.
#

#
# Required parameter
# =====
#
# $1 = Hostname of Host machine (ComunicacionsXX)
#

#
# INPUT PARAMETER CHECK
# -----
#

# Define how has to be hostname (defined in Required parameter
# section)
# using regular expressions
hostHostNameRegex1="^Comunicacions0[1-9]{1}$"
hostHostNameRegex2="^Comunicacions1[0-8]{1}$"

# Check if user has entered a correct hostname for Host machine
if [[ ! ( $1 =~ $hostHostNameRegex1 || $1 =~ $hostHostNameRegex2 ) ]];
then
    error="El hostname de la maquina real no es correcte."
    echo $error
    echo $error >> setControllerParams.log
    exit 1
fi

#
# SET THE COMMON VARIABLES TO WORK WITH
# -----

```

```

#
hostInfo=$(cat hosts | grep $1)

# Hostname that corresponds to the VM
hostName=$(echo $hostInfo | awk '{ print $2 }')

# Management IP that corresponds to $1
managementIP=$(echo $hostInfo | awk '{ print $1 }')

# Tunnel IP that corresponds to $1. I achieve this
# replacing the 6th character with a '2' with sed command
tunnelIP=$(echo $managementIP | sed s/./2/6)

# Path to interfaces configuration file
interfaces="/etc/network/interfaces"

# Path to nova configuration file
novaConf="/etc/nova/nova.conf"

# Path to ml2 plugin configuration file
ml2Conf="/etc/neutron/plugins/ml2/ml2_conf.ini"

#
# MAKE COMMON SETTINGS
# -----
#
# Set IP for Management net
sed -i '/address 10.1.1/c\address '$managementIP'' '$interfaces'

# Change hostname
echo $hostName > /etc/hostname
# Change /etc/hosts with the template
cat hosts > /etc/hosts

#
# MAKE THE SPECIFIC SETTINGS
# -----
#
if [[ $hostName == "Controller" ]]; then
# CONTROLLER node
# Set IP for External net
externalIP=$(echo $hostInfo | awk '{ print $5 }')
sed -i '/address 84.88.155/c\address '$externalIP'' '$interfaces'

# MariaDB settings
mariaDBConf="/etc/mysql/my.cnf"
sed -i '/bind-address =/c\bind-address = '$managementIP''
$mariaDBConf

# Nova service settings
# Set my_ip parameter in $novaconf
sed -i '/my_ip =/c\my_ip = '$managementIP'' '$novaConf'
# Set vncserver_listen parameter in $novaConf
sed -i '/vncserver_listen =/c\vncserver_listen =
'$managementIP'' '$novaConf'
# Set vncserver_proxyclient_address parameter in $novaConf

```

```

    sed -i '/vncserver_proxycient_address
=/c\vncserver_proxycient_address = '$managementIP' '$novaConf

    # Ceilometer service settings
    ceilometerConf="/etc/mongodb.conf"
    sed -i '/bind_ip =/c\bind_ip = '$managementIP' '$ceilometerConf

elif [[ $hostName == "Network" ]]; then
    # NETWORK node
    # Set IP for Tunnel net
    sed -i '/address 10.1.2/c\address '$tunnelIP' '$interfaces

    # Neutron service settings
    # Set local_ip parameter in $ml2Conf
    sed -i '/local_ip = /c\local_ip = '$tunnelIP' '$ml2Conf
else
    # COMPUTE nodes
    # Set IP for Tunnel net
    sed -i '/address 10.1.2/c\address '$tunnelIP' '$interfaces

    # Nova service settings
    # Set my_ip parameter in $novaconf
    sed -i '/my_ip =/c\my_ip = '$managementIP' '$novaConf
    # Set vncserver_proxycient_address parameter in $novaConf
    sed -i '/vncserver_proxycient_address
=/c\vncserver_proxycient_address = '$managementIP' '$novaConf

    # Neutron service settings
    # Set local_ip parameter in $ml2Conf
    sed -i '/local_ip = /c\local_ip = '$tunnelIP' '$ml2Conf
fi

# Reboot the machine to make changes get active
reboot

```

### 14.3 Fixter d' script cloudAccess.bat

```

echo                84.88.155.108                controller                >>
"C:\Windows\System32\drivers\etc\hosts"

```

## 15. Manual d'usuari i/o instal·lació

Els usuaris d'aquest *cloud* estaran connectats a la xarxa local de la Universitat, així que el primer pas és afegir l'adreça pública del node *Controller* al seu fitxer de *hosts*. Per fer-ho només cal que executin l'*script* presentat als annexos, *cloudAccess.bat* com a administradors.

Un cop l'ordinador d'accés conegui l'associació entre el nom "controller" i la seva IP, a la barra d'adreces de qualsevol navegador només cal posar "controller/horizon", i si l'administrador del *cloud* els ha subministrat unes credencials, només caldrà posar-les:

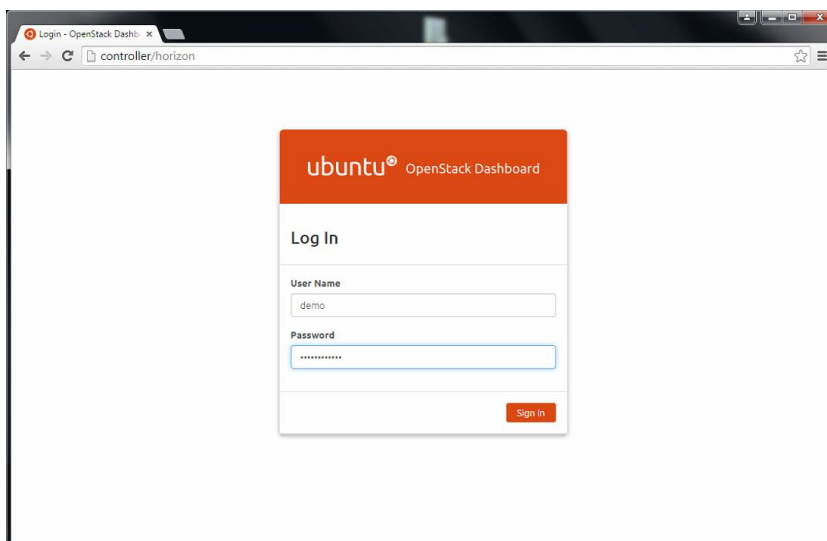


Figura 116: Pantalla de login

Un cop a dins ja poden treballar amb els seus recursos assignats i administrar-los:

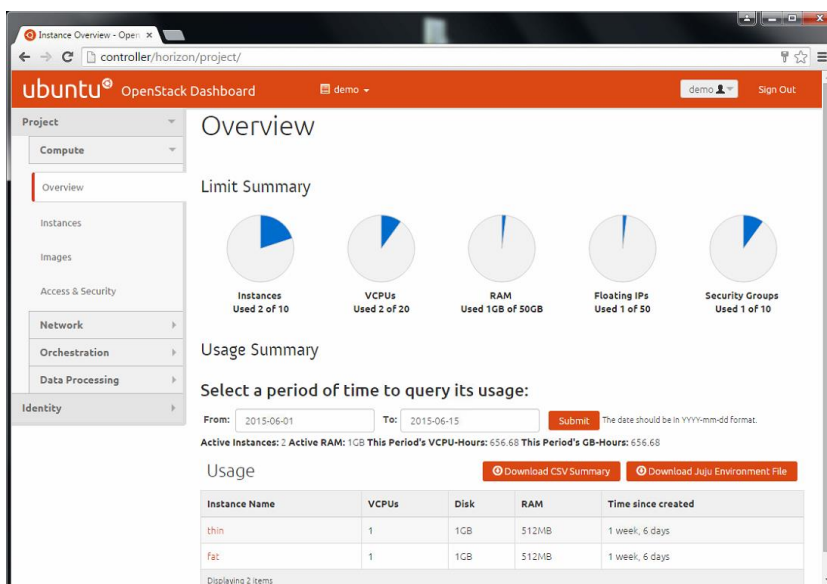


Figura 117: Vista dels recursos disponibles de l'usuari



Per més informació, només cal visitar la guia d'usuari que OpenStack ens proporciona [aquí](#)<sup>23</sup>.

---

<sup>23</sup> <http://docs.openstack.org/user-guide/index.html>