Proyecto Final de Máster en Software Libre

“Estado del arte en soluciones de virtualización. Sistemas gestores de Cloud: OpenNebula”

Anexo2: Comandos ejecutados junto con sus salidas correspondientes, para realizar la instalación y puesta en servicio de la solución en el entorno de producción.

Administración de redes y sistemas operativos

Autor: José Antonio Montes Serena

Consultor: Jordi Massaguer Pla

Tutor externo: Antonio Rodil Garrido

Empresa colaboradora: ARTIC S.L.

Fecha: Diciembre de 2013

Licencia:

Resumen del Anexo2:

En el presente documento se pretende recoger todos los comandos ejecutados junto con sus salidas correspondientes, acompañados de los comentarios necesarios, para seguir los pasos acometidos durante los procesos de instalación, pruebas, y puesta a punto de la solución de producción, de gestión del datacenter virtualizado basado en OpenNebula. Después de realizar todos los pasos recogidos en el presente documento, la plataforma ha quedado preparada para producción. Se ha intentado ser lo más riguroso posible (incluyendo los problemas encontrados y las soluciones adoptadas), de tal forma que pueda seguir de guía a cualquiera que desee implantar la misma solución, si bien hay que tener en cuenta que los recursos físicos disponibles pueden cambiar de una instalación a otra (no existen dos entornos de producción idénticos), así como las versiones de software de los paquetes utilizados.

Los pasos seguidos durante la instalación y puesta en servicio de la plataforma se han basado en aquellos realizados durante la instalación y pruebas de la maqueta, con lo que se han obviado algunos pasos idénticos ya mostrados en el anexo correspondiente a la maqueta (como la instalación de la distribución del sistema operativo), por no aportar información adicional al proceso de instalación de la plataforma de producción.

Por último, comentar que los pasos recogidos en el presente documento siguen estrictamente el mismo orden que el abordado en la ejecución de los comandos realizados durante la implantación de la plataforma. Se podría decir que corresponde más bien a un *log* personal de los comandos y salidas ejecutados, acompañado de los comentarios necesarios intercalados, todo ello dentro de un contexto eminentemente práctico de registrar lo que se ha hecho para conseguir desplegar la solución.

Este anexo (junto con el anexo 1, correspondiente a la instalación del entorno de pruebas) forma parte de los materiales entregables del Trabajo de Fin de Máster presentado por el alumno.

Índice de contenidos

1. Instalación del software en los servidores.....................................5
2. Configuración inicial de OpenNebula..15
2.1. Configuración de las cuentas y permisos del sistema.............15
2.2. Preparación de las redes en los hosts......................................17
2.3. Configuración del servicio NFS..24
2.4. Configuración del 802.1Q para aislar las VLANs.......................26
2.5. Primeros pasos con la configuración de OpenNebula.............32
2.6. Definiendo las primeras platillas..35
2.7. Realizando pruebas de migración en vivo de las instancias....47
2.8. Optimizando la instanciación de las imágenes en los DS.........50
3. Migración de los host a direccionamiento público......................52
4. Preparación de la plataforma para pruebas y producción..........68
4.1. Trabajando con el datastore qcow y preparando las imágenes69
4.2. Preparando las redes de uso para los usuarios........................81
4.3. Trabajando con cuentas de usuario..83
4.3.1. Trabajando con el router virtual para el usuario customer1. .85
4.3.2. Recuperando la imagen tty_linux...110
4.3.3. Preparando plantillas de prueba para customer1 120
4.3.4. Trabajando con la persistencia de las imágenes..................122
4.3.5. Configurando una red privada para el usuario customer1...126
4.3.6. Preparando el resto de plantillas para customer1................127
4.4. Modificando el scheduler para pruebas de rendimiento.........131
4.4.1. Preparación de las pruebas de rendimiento.........................133
5. Instalación de Ganglia como sistema de monitorización..........143
6. Creación de un usuario adicional..148
6. Asignación de cuotas de uso a los usuarios..............................160
7. Últimos pasos de configuración de la plataforma......................164

Comandos y salidas ejecutados y comentados, realizados en en entorno de producción

1. Instalación del software en los servidores

Partimos de la instalación base del sistema operativo, CentOS 6.4. y seguimos exactamente los mismos pasos que con la instalación del SO en todas las máquinas utilizadas en la maqueta (véase los pasos seguidos en el anexo1 del presente TFM).

Dejamos instalados con el sistema operativo los host one-node1 y one-node2. Dos servidores exactamente iguales. A continuación procedemos con la instalación de los paquetes de OpenNebula.

Primero instalamos el repositorio EPEL en ambas máquinas, y además creamos el fichero para los repositorios actualizados de OpenNebula:

```
[root@one-node1 yum.repos.d]# more opennebula.repo
[opennebula]
name=opennebula
baseurl=http://opennebula.org/repo/CentOS/6/stable/x86_64
enabled=1
gpgcheck=0

[root@one-node2 yum.repos.d]# more opennebula.repo
[opennebula]
name=opennebula
baseurl=http://opennebula.org/repo/CentOS/6/stable/x86_64
enabled=1
gpgcheck=0
```

A continuación instalamos el repositorio EPEL:

```
[root@one-node1 ~]# wget http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
--2013-11-13 00:43:03--  http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
Resolving dl.fedoraproject.org... 209.132.181.25, 209.132.181.26, 209.132.181.27, ...
Connecting to dl.fedoraproject.org|209.132.181.25|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14540 (14K) [application/x-rpm]
Saving to: “epel-release-6-8.noarch.rpm”

100%
[====================================================================================================
=====>] 14,540      84.9K/s   in 0.2s

2013-11-13 00:43:05 (84.9 KB/s) - “epel-release-6-8.noarch.rpm” saved [14540/14540]

[root@one-node1 ~]# yum localinstall epel-release-6-8.noarch.rpm
Loaded plugins: fastestmirror, security
```

```
Setting up Local Package Process
Examining epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
Marking epel-release-6-8.noarch.rpm to be installed
Loading mirror speeds from cached hostfile
 * base: centos.mirror.privado1.es
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
opennebula
| 2.9 kB     00:00
opennebula/primary_db
|  16 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:6-8 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================
==============================================
 Package                               Arch                          Version
Repository                                      Size
=====================================================================================================
==============================================
Installing:
 epel-release                          noarch                        6-8                         /epel-
release-6-8.noarch                        22 k

Transaction Summary
=====================================================================================================
==============================================
Install       1 Package(s)

Total size: 22 k
Installed size: 22 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : epel-release-6-8.noarch
1/1
  Verifying  : epel-release-6-8.noarch
1/1

Installed:
  epel-release.noarch 0:6-8
```

```
Complete!

ot@one-node1 ~]# yum repolist
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
epel/metalink
|  24 kB     00:00
 * base: centos.mirror.privado1.es
 * epel: ftp.cica.es
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
epel
| 4.2 kB     00:00
epel/primary_db
| 5.7 MB     00:00
repo id                                                                     repo name
status
base                                                                        CentOS-6 - Base
6,381
epel                                                                        Extra Packages for Enterprise
Linux 6 - x86_64                                                                   9,984
extras                                                                      CentOS-6 - Extras
13
opennebula                                                                  opennebula
21
updates                                                                     CentOS-6 - Updates
1,530
repolist: 17,929
```

Bien, hacemos lo mismo con el otro nodo:

```
[root@one-node2 ~]# wget http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
--2013-11-13 00:46:07--  http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
Resolving dl.fedoraproject.org... 209.132.181.24, 209.132.181.25, 209.132.181.26, ...
Connecting to dl.fedoraproject.org|209.132.181.24|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14540 (14K) [application/x-rpm]
Saving to: "epel-release-6-8.noarch.rpm"

100%[=============================================================================================>]
14,540      85.0K/s   in 0.2s

2013-11-13 00:46:08 (85.0 KB/s) - "epel-release-6-8.noarch.rpm" saved [14540/14540]

[root@one-node2 ~]#  yum localinstall epel-release-6-8.noarch.rpm
Loaded plugins: fastestmirror, security
Setting up Local Package Process
Examining epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
```

```
Marking epel-release-6-8.noarch.rpm to be installed
Loading mirror speeds from cached hostfile
 * base: centos.mirror.privado1.es
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
opennebula
| 2.9 kB     00:00
opennebula/primary_db
|  16 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:6-8 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

====================================================================================================
===================================
 Package                          Arch                         Version                  Repository
Size
====================================================================================================
===================================
Installing:
 epel-release                     noarch                       6-8                      /epel-release-6-
8.noarch                      22 k

Transaction Summary
====================================================================================================
===================================
Install       1 Package(s)

Total size: 22 k
Installed size: 22 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : epel-release-6-8.noarch
1/1
  Verifying  : epel-release-6-8.noarch
1/1

Installed:
  epel-release.noarch 0:6-8

Complete!
```

```
[root@one-node2 ~]# yum repolist
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
epel/metalink
|  24 kB     00:00
 * base: centos.mirror.privado1.es
 * epel: mir01.syntis.net
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
epel
| 4.2 kB     00:00
epel/primary_db
| 5.7 MB     00:00
repo id                                         repo name
status
base                                            CentOS-6 - Base
6,381
epel                                            Extra Packages for Enterprise Linux 6 - x86_64
9,984
extras                                          CentOS-6 - Extras
13
opennebula                                      opennebula
21
updates                                         CentOS-6 - Updates
1,530
repolist: 17,929
```

Bien, ahora en el nodo 1 instalamos todos los paquetes de administración, mientras que en el nodo 2 sólo instalamos los correspondientes al KVM:

```
[root@one-node1 ~]# yum install opennebula-node-kvm opennebula-server opennebula-sunstone opennebula-
gate
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
 * base: centos.mirror.privado1.es
 * epel: ftp.cica.es
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
Setting up Install Process
Dependencies Resolved

=====================================================================================================
================================================================================
 Package                                          Arch                              Version
Repository                            Size
=====================================================================================================
================================================================================
Installing:
 opennebula-gate                                  x86_64                            4.2.0-1
opennebula                           8.9 k
```

```
 opennebula-node-kvm                             x86_64                           4.2.0-1
opennebula                            6.5 k
 opennebula-server                               x86_64                           4.2.0-1
opennebula                            864 k
 opennebula-sunstone                             x86_64                           4.2.0-1
opennebula                            1.1 M
Installing for dependencies:
 celt051                                         x86_64                           0.5.1.3-0.el6
base                                   50 k
 compat-readline5                                x86_64                           5.2-17.1.el6
base                                  130 k
 cyrus-sasl-md5                                  x86_64                           2.1.23-13.el6_3.1
base                                   47 k
 dnsmasq                                         x86_64                           2.48-13.el6
base                                  149 k
 ebtables                                        x86_64                           2.0.9-6.el6
base                                   95 k
 flac                                            x86_64                           1.2.1-6.1.el6
base                                  243 k
 genisoimage                                     x86_64                           1.1.9-12.el6
base                                  348 k
 gnutls-utils                                    x86_64                           2.8.5-10.el6_4.2
updates                               100 k
 gpxe-roms-qemu                                  noarch                           0.9.7-6.9.el6
base                                  219 k
 iscsi-initiator-utils                           x86_64                           6.2.0.873-2.el6
base                                  655 k
 libasyncns                                      x86_64                           0.8-1.1.el6
base                                   24 k
 libsndfile                                      x86_64                           1.0.20-5.el6
base                                  233 k
 libvirt                                         x86_64                           0.10.2-18.el6_4.14
updates                               2.3 M
 libvirt-client                                  x86_64                           0.10.2-18.el6_4.14
updates                               4.0 M
 log4cpp                                         x86_64                           1.0-13.el6
base                                  537 k
 lzo                                             x86_64                           2.03-3.1.el6
base                                   55 k
 lzop                                            x86_64                           1.02-0.9.rc1.el6
base                                   50 k
 nc                                              x86_64                           1.84-22.el6
base                                   57 k
 netcf-libs                                      x86_64                           0.1.9-3.el6
base                                   51 k
 numad                                           x86_64                           0.5-
8.20121015git.el6                                   base                                 27 k
 opennebula                                      x86_64                           4.2.0-1
opennebula                             58 k
 opennebula-common                               x86_64                           4.2.0-1
opennebula                            5.4 k
 opennebula-ruby                                 x86_64                           4.2.0-1
opennebula                             53 k
```

```
 pulseaudio-libs                                  x86_64                           0.9.21-14.el6_3
base                                 462 k
 qemu-img                                         x86_64                           2:0.12.1.2-
2.355.0.1.el6_4.9                            updates                               475 k
 qemu-kvm                                         x86_64                           2:0.12.1.2-
2.355.0.1.el6_4.9                            updates                               1.3 M
 radvd                                            x86_64                           1.6-1.el6
base                                  75 k
 ruby                                             x86_64                           1.8.7.352-12.el6_4
updates                              534 k
 ruby-irb                                         x86_64                           1.8.7.352-12.el6_4
updates                              313 k
 ruby-libs                                        x86_64                           1.8.7.352-12.el6_4
updates                              1.6 M
 ruby-rdoc                                        x86_64                           1.8.7.352-12.el6_4
updates                              376 k
 rubygem-daemons                                  noarch                           1.0.10-2.el6
epel                                 122 k
 rubygem-eventmachine                             x86_64                           0.12.10-4.el6
epel                                 355 k
 rubygem-json                                     x86_64                           1.4.6-1.el6
epel                                 457 k
 rubygem-nokogiri                                 x86_64                           1.4.3.1-1.el6
epel                                 308 k
 rubygem-rack                                     noarch                           1:1.1.0-2.el6
epel                                 446 k
 rubygem-rack-test                                noarch                           0.5.4-1.el6
epel                                  62 k
 rubygem-sequel                                   noarch                           4.1.1-1.el6
epel                                 2.2 M
 rubygem-sinatra                                  noarch                           1:1.0-2.el6
epel                                 306 k
 rubygem-sqlite3-ruby                             x86_64                           1.2.4-5.el6
epel                                 221 k
 rubygem-thin                                     x86_64                           1.2.8-4.el6
epel                                 187 k
 rubygem-uuidtools                                noarch                           2.1.1-1.el6
epel                                  30 k
 rubygems                                         noarch                           1.3.7-4.el6_4
updates                              207 k
 seabios                                          x86_64                           0.6.1.2-26.el6
base                                  91 k
 sgabios-bin                                      noarch                           0-
0.3.20110621svn.el6                                   base                                  6.6 k
 spice-server                                     x86_64                           0.12.0-12.el6_4.5
updates                              326 k
 usbredir                                         x86_64                           0.5.1-1.el6
base                                  40 k
 vgabios                                          noarch                           0.6b-3.7.el6
base                                  42 k
 xmlrpc-c-c++                                     x86_64                           1.16.24-
1209.1840.el6                                   base                                   66 k
```

```
 xmlrpc-c-client++                               x86_64                            1.16.24-
1209.1840.el6                                   base                                  36 k
 yajl                                            x86_64                            1.0.7-3.el6
base                                  27 k


Transaction Summary
====================================================================================================
================================================================================
Install      56 Package(s)


Total download size: 22 M
Installed size: 73 M
Is this ok [y/N]: y





Installed:
  opennebula-gate.x86_64 0:4.2.0-1 opennebula-node-kvm.x86_64 0:4.2.0-1 opennebula-server.x86_64
0:4.2.0-1 opennebula-sunstone.x86_64 0:4.2.0-1


Dependency Installed:
  celt051.x86_64 0:0.5.1.3-0.el6               compat-readline5.x86_64 0:5.2-17.1.el6
cyrus-sasl-md5.x86_64 0:2.1.23-13.el6_3.1 dnsmasq.x86_64 0:2.48-13.el6
  ebtables.x86_64 0:2.0.9-6.el6                flac.x86_64 0:1.2.1-6.1.el6
genisoimage.x86_64 0:1.1.9-12.el6          gnutls-utils.x86_64 0:2.8.5-10.el6_4.2
  gpxe-roms-qemu.noarch 0:0.9.7-6.9.el6        iscsi-initiator-utils.x86_64 0:6.2.0.873-2.el6
libasyncns.x86_64 0:0.8-1.1.el6            libsndfile.x86_64 0:1.0.20-5.el6
  libvirt.x86_64 0:0.10.2-18.el6_4.14          libvirt-client.x86_64 0:0.10.2-18.el6_4.14
log4cpp.x86_64 0:1.0-13.el6                lzo.x86_64 0:2.03-3.1.el6
  lzop.x86_64 0:1.02-0.9.rc1.el6               nc.x86_64 0:1.84-22.el6
netcf-libs.x86_64 0:0.1.9-3.el6            numad.x86_64 0:0.5-8.20121015git.el6
  opennebula.x86_64 0:4.2.0-1                  opennebula-common.x86_64 0:4.2.0-1
opennebula-ruby.x86_64 0:4.2.0-1           pulseaudio-libs.x86_64 0:0.9.21-14.el6_3
  qemu-img.x86_64 2:0.12.1.2-2.355.0.1.el6_4.9 qemu-kvm.x86_64 2:0.12.1.2-2.355.0.1.el6_4.9
radvd.x86_64 0:1.6-1.el6                   ruby.x86_64 0:1.8.7.352-12.el6_4
  ruby-irb.x86_64 0:1.8.7.352-12.el6_4         ruby-libs.x86_64 0:1.8.7.352-12.el6_4          ruby-
rdoc.x86_64 0:1.8.7.352-12.el6_4     rubygem-daemons.noarch 0:1.0.10-2.el6
  rubygem-eventmachine.x86_64 0:0.12.10-4.el6  rubygem-json.x86_64 0:1.4.6-1.el6
rubygem-nokogiri.x86_64 0:1.4.3.1-1.el6    rubygem-rack.noarch 1:1.1.0-2.el6
  rubygem-rack-test.noarch 0:0.5.4-1.el6       rubygem-sequel.noarch 0:4.1.1-1.el6
rubygem-sinatra.noarch 1:1.0-2.el6         rubygem-sqlite3-ruby.x86_64 0:1.2.4-5.el6
  rubygem-thin.x86_64 0:1.2.8-4.el6            rubygem-uuidtools.noarch 0:2.1.1-1.el6
rubygems.noarch 0:1.3.7-4.el6_4            seabios.x86_64 0:0.6.1.2-26.el6
  sgabios-bin.noarch 0:0-0.3.20110621svn.el6   spice-server.x86_64 0:0.12.0-12.el6_4.5
usbredir.x86_64 0:0.5.1-1.el6              vgabios.noarch 0:0.6b-3.7.el6
  xmlrpc-c-c++.x86_64 0:1.16.24-1209.1840.el6  xmlrpc-c-client++.x86_64 0:1.16.24-1209.1840.el6
yajl.x86_64 0:1.0.7-3.el6


Complete!
[root@one-node1 ~]#
```

En el nodo2 sólo tenemos que instalar el módulo correspondiente al KVM:

```
[root@one-node2 ~]# yum install opennebula-node-kvm
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
 * base: centos.mirror.privado1.es
 * epel: ftp.cica.es
 * extras: centos.mirror.privado1.es
 * updates: centos.mirror.privado1.es
Setting up Install Process
Resolving Dependencies

Dependencies Resolved

====================================================================================================
===================================
 Package                                     Arch                     Version
Repository                     Size
====================================================================================================
===================================
Installing:
 opennebula-node-kvm                         x86_64                   4.2.0-1
opennebula                    6.5 k
Installing for dependencies:
 celt051                                     x86_64                   0.5.1.3-0.el6
base                           50 k
 compat-readline5                            x86_64                   5.2-17.1.el6
base                          130 k
 cyrus-sasl-md5                              x86_64                   2.1.23-13.el6_3.1
base                           47 k
 dnsmasq                                     x86_64                   2.48-13.el6
base                          149 k
 ebtables                                    x86_64                   2.0.9-6.el6
base                           95 k
 flac                                        x86_64                   1.2.1-6.1.el6
base                          243 k
 gnutls-utils                                x86_64                   2.8.5-10.el6_4.2
updates                       100 k
 gpxe-roms-qemu                              noarch                   0.9.7-6.9.el6
base                          219 k
 iscsi-initiator-utils                       x86_64                   6.2.0.873-2.el6
base                          655 k
 libasyncns                                  x86_64                   0.8-1.1.el6
base                           24 k
 libsndfile                                  x86_64                   1.0.20-5.el6
base                          233 k
 libvirt                                     x86_64                   0.10.2-18.el6_4.14
updates                       2.3 M
 libvirt-client                              x86_64                   0.10.2-18.el6_4.14
updates                       4.0 M
```

```
 lzo                                  x86_64                2.03-3.1.el6
base                       55 k
 lzop                                 x86_64                1.02-0.9.rc1.el6
base                       50 k
 nc                                   x86_64                1.84-22.el6
base                       57 k
 netcf-libs                           x86_64                0.1.9-3.el6
base                       51 k
 numad                                x86_64                0.5-8.20121015git.el6
base                       27 k
 opennebula-common                    x86_64                4.2.0-1
opennebula                5.4 k
 pulseaudio-libs                      x86_64                0.9.21-14.el6_3
base                      462 k
 qemu-img                             x86_64                2:0.12.1.2-2.355.0.1.el6_4.9
updates                   475 k
 qemu-kvm                             x86_64                2:0.12.1.2-2.355.0.1.el6_4.9
updates                   1.3 M
 radvd                                x86_64                1.6-1.el6
base                       75 k
 ruby                                 x86_64                1.8.7.352-12.el6_4
updates                   534 k
 ruby-libs                            x86_64                1.8.7.352-12.el6_4
updates                   1.6 M
 seabios                              x86_64                0.6.1.2-26.el6
base                       91 k
 sgabios-bin                          noarch                0-0.3.20110621svn.el6
base                      6.6 k
 spice-server                         x86_64                0.12.0-12.el6_4.5
updates                   326 k
 usbredir                             x86_64                0.5.1-1.el6
base                       40 k
 vgabios                              noarch                0.6b-3.7.el6
base                       42 k
 yajl                                 x86_64                1.0.7-3.el6
base                       27 k


Transaction Summary
=====================================================================================================
===================================
Install      32 Package(s)


Total download size: 13 M
Installed size: 43 M
Is this ok [y/N]:y


Installed:
  opennebula-node-kvm.x86_64 0:4.2.0-1


Dependency Installed:
  celt051.x86_64 0:0.5.1.3-0.el6                compat-readline5.x86_64 0:5.2-17.1.el6   cyrus-sasl-
md5.x86_64 0:2.1.23-13.el6_3.1
  dnsmasq.x86_64 0:2.48-13.el6                  ebtables.x86_64 0:2.0.9-6.el6            flac.x86_64
0:1.2.1-6.1.el6
  gnutls-utils.x86_64 0:2.8.5-10.el6_4.2        gpxe-roms-qemu.noarch 0:0.9.7-6.9.el6    iscsi-
initiator-utils.x86_64 0:6.2.0.873-2.el6
  libasyncns.x86_64 0:0.8-1.1.el6               libsndfile.x86_64 0:1.0.20-5.el6
libvirt.x86_64 0:0.10.2-18.el6_4.14
  libvirt-client.x86_64 0:0.10.2-18.el6_4.14   lzo.x86_64 0:2.03-3.1.el6                lzop.x86_64
0:1.02-0.9.rc1.el6
  nc.x86_64 0:1.84-22.el6                       netcf-libs.x86_64 0:0.1.9-3.el6          numad.x86_64
0:0.5-8.20121015git.el6
  opennebula-common.x86_64 0:4.2.0-1            pulseaudio-libs.x86_64 0:0.9.21-14.el6_3 qemu-
img.x86_64 2:0.12.1.2-2.355.0.1.el6_4.9
  qemu-kvm.x86_64 2:0.12.1.2-2.355.0.1.el6_4.9 radvd.x86_64 0:1.6-1.el6                 ruby.x86_64
0:1.8.7.352-12.el6_4
  ruby-libs.x86_64 0:1.8.7.352-12.el6_4         seabios.x86_64 0:0.6.1.2-26.el6          sgabios-
bin.noarch 0:0-0.3.20110621svn.el6
  spice-server.x86_64 0:0.12.0-12.el6_4.5       usbredir.x86_64 0:0.5.1-1.el6
vgabios.noarch 0:0.6b-3.7.el6
  yajl.x86_64 0:1.0.7-3.el6


Complete!
[root@one-node2 ~]#
```

Ahora comenzamos con la configuración de OpenNebula.

2. Configuración inicial de OpenNebula

2.1. Configuración de las cuentas y permisos del sistema

En los hosts tenemos que añadir el grupo wheel al usuario oneadmin para que tenga privilegios al trabajar con la red, etc...

```
[root@one-node1 ~]# gpasswd -a oneadmin wheel
Adding user oneadmin to group wheel
[root@one-node1 ~]#

[root@one-node2 ~]# gpasswd -a oneadmin wheel
Adding user oneadmin to group wheel
[root@one-node2 ~]#
```

Ahora definimos una password para el usuario oneadmin en los dos hosts:

```
[root@one-node1 ~]# passwd oneadmin
Changing password for user oneadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@one-node1 ~]#

[root@one-node2 ~]# passwd oneadmin
```

```
Changing password for user oneadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@one-node2 ~]#
```

Nos aseguramos que podemos entrar con el usuario oneadmin de forma automática:

```
[root@one-node1 ~]# su - oneadmin
[oneadmin@one-node1 ~]$ ssh one-node1
The authenticity of host 'one-node1 (192.168.25.203)' can't be established.
RSA key fingerprint is 51:9d:de:32:55:55:eb:e7:01:19:76:5c:a2:7a:c0:a4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'one-node1,192.168.25.203' (RSA) to the list of known hosts.
[oneadmin@one-node1 ~]$
[oneadmin@one-node1 ~]$ scp -Crp .ssh oneadmin@one-node2:.
The authenticity of host 'one-node2 (192.168.25.204)' can't be established.
RSA key fingerprint is a0:f6:fa:51:29:ad:1c:7b:0a:86:bf:80:88:b1:5f:79.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'one-node2,192.168.25.204' (RSA) to the list of known hosts.
oneadmin@one-node2's password:
id_dsa.pub
100%  608     0.6KB/s   00:00
authorized_keys
100%  608     0.6KB/s   00:00
known_hosts
100%  812     0.8KB/s   00:00
id_dsa
100%  672     0.7KB/s   00:00
[oneadmin@one-node1 ~]$ ssh oneadmin@one-node2
Last login: Wed Nov 13 01:54:22 2013 from one-node2
[oneadmin@one-node2 ~]$ ssh oneadmin@one-node2
Last login: Wed Nov 13 01:59:15 2013 from one-node1
[oneadmin@one-node2 ~]$ ssh oneadmin@one-node1
Last login: Wed Nov 13 01:51:37 2013 from one-node1
[oneadmin@one-node1 ~]$
```

Preparamos la password inicial que tendrá el usuario oneadmin al entrar por primera vez en el OpenNebula:

```
[oneadmin@one-node1 ~]$ more .one/one_auth
oneadmin:adivinalo
```

En el frontend tenemos que preparar el sunstone para poder acceder desde fuera.

En **/etc/one/sunstone-server.conf** cambiamos lo siguiente:

```
[root@one-node1 one]# diff sunstone-server.conf sunstone-server.conf.org
31c31
< :host: 0.0.0.0
---
> :host: 127.0.0.1
```

2.2. Preparación de las redes en los hosts

Vamos a trabajar con las redes que utilizaremos en nuestro sistema en producción.

Los dos hosts tendrán configurados cuatro redes físicas diferentes:

- em1: red de salida al exterior para las máquinas físicas. También será la red de gestión del interfaz web. En esta solución se hará con acceso público.
- em2: red de salida al exterior con IPs públicas para las máquinas virtuales. No todas las máquinas virtuales necesitarán de una IP pública para funcionar.
- em3: red de gestión y servicio de OpenNebula. Es una red privada exclusiva para los dos hosts, sin acceso desde el exterior ni por parte de las máquinas virtuales.
- em4: Es la red donde se definirán las VLANs por 802.1Q privadas de las máquinas virtuales, para conexión directa entre ellas con segmentos privados de red, sin acceso directo al exterior.

Además contaremos con el bridge virbr0 que ofrece KVM por defecto, para facilitar el acceso al exterior de las máquinas virtuales mediante NAT, aunque en una fase posterior la deshabilitaremos, para dejar ese papel a las instancias de los routers virtuales.

Primero configuraremos el bridge de KVM apropiadamente. Como vamos a contar con un número grande de máquinas virtuales en nuestra nube privada, permitiendo la migración de máquinas virtuales entre los hosts, vamos a definir una red de NAT lo suficientemente grande (optamos con un /21):

```
Network:   192.168.120.0/21     11000000.10101000.01111 000.00000000
Netmask:   255.255.248.0 = 21   11111111.11111111.11111 000.00000000
HostMin:   192.168.120.1        11000000.10101000.01111 000.00000001
HostMax:   192.168.127.254      11000000.10101000.01111 111.11111110
Broadcast: 192.168.127.255      11000000.10101000.01111 111.11111111
Hosts/Net: 2046
```

Para ello tendremos que editar los ficheros de configuración de KVM en ambos hosts, cambiando la red definida por defecto (192.168.122.0/24) por la nueva.

Ello se consigue modificando el fichero:

```
/var/lib/libvirt/network/default.xml
```

Paramos el servicio libvirtd:

```
[root@one-node1 ~]# service libvirtd status
libvirtd (pid  4836) is running...
[root@one-node1 ~]# service libvirtd stop
Stopping libvirtd daemon:                                  [  OK  ]
[root@one-node1 ~]#
```

Paramos también las iptables:

```
[root@one-node1 ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere            udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere            udp dpt:bootps
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:bootps
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  10.60.1.0/24         anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:9869
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:29876
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             192.168.122.0/24    state RELATED,ESTABLISHED
ACCEPT     all  --  192.168.122.0/24     anywhere
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             anywhere            reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere            reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@one-node1 ~]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: nat mangle filte[  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@one-node1 ~]#
```

Editamos el fichero /var/lib/libvirt/network/default.xml:

```
[root@one-node1 network]# diff default.xml default.xml.org
14c14
<   <ip address='192.168.120.1' netmask='255.255.248.0'>
---
>   <ip address='192.168.122.1' netmask='255.255.255.0'>
16c16
<       <range start='192.168.120.2' end='192.168.127.254' />
---
>       <range start='192.168.122.2' end='192.168.122.254' />
[root@one-node1 network]#
```

Arrancamos el servicio de nuevo:

```
[root@one-node1 network]# service libvirtd start
Starting libvirtd daemon:                                  [  OK  ]
[root@one-node1 network]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere             udp dpt:bootps
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:bootps

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             192.168.120.0/21    state RELATED,ESTABLISHED
ACCEPT     all  --  192.168.120.0/21     anywhere
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             anywhere             reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere             reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Bien, vemos que ese fichero se utiliza para establecer las reglas fordwarding en las iptables, pero no se utiliza para configurar el interfaz de red virbr0 en KVM:

```
[root@one-node1 network]# ifconfig virbr0
virbr0    Link encap:Ethernet  HWaddr 52:54:00:91:7E:04
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Esto lo conseguimos con el siguiente comando:

```
[root@one-node1 network]# virsh net-edit default
<network>
  <name>default</name>
  <uuid>c93540ed-c9fa-4079-a4c7-2e75ca0ad673</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:91:7E:04'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
  </ip>
</network>
```

Y después de editarlo lo dejamos así:

```
<network>
  <name>default</name>
  <uuid>c93540ed-c9fa-4079-a4c7-2e75ca0ad673</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:91:7E:04'/>
  <ip address='192.168.120.1' netmask='255.255.248.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.127.254' />
    </dhcp>
  </ip>
</network>
Network default XML configuration edited.

[root@one-node1 network]# virsh net-destroy default
Network default destroyed

[root@one-node1 network]# ifconfig virbr0
virbr0: error fetching interface information: Device not found
[root@one-node1 network]# virsh net-start default
Network default started

[root@one-node1 network]# ifconfig virbr0
virbr0    Link encap:Ethernet  HWaddr 52:54:00:91:7E:04
          inet addr:192.168.120.1  Bcast:192.168.127.255  Mask:255.255.248.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@one-node1 network]#
```

Hacemos lo mismo con el otro host. Nota: el utilizar la misma red en ambos hosts nos permitirá realizar las migraciones en vivo de las máquinas virtuales entre los hosts.

Por otra parte, para evitar IPs repetidas en máquinas virtuales situadas en diferentes hosts, realizaremos la asignación de la IP en cada máquina virtual mediante contextualización, utilizando una plantilla de red.

Esta vez lo hacemos de otra forma:

```
[root@one-node2 ~]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: nat mangle filte[  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@one-node2 ~]# virsh net-destroy default
Network default destroyed
```

```
[root@one-node2 ~]# virsh net-edit default
<network>
  <name>default</name>
  <uuid>c3459eaa-ec6e-428c-8c45-97906d71853c</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:7B:BA:7B'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
  </ip>
</network>
```

Y lo cambiamos por esto:

```
<network>
  <name>default</name>
  <uuid>c3459eaa-ec6e-428c-8c45-97906d71853c</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:7B:BA:7B'/>
  <ip address='192.168.120.1' netmask='255.255.248.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.127.254' />
    </dhcp>
  </ip>
</network>
Network default XML configuration edited.

[root@one-node2 ~]# virsh net-start default
Network default started

[root@one-node2 ~]# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

```
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.120.0/21 -o virbr0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.120.0/21 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
[root@one-node2 ~]#
```

Comprobamos el contenido del fichero de configuración por defecto:

```
[root@one-node2 ~]# more /var/lib/libvirt/network/default.xml
<!--
WARNING: THIS IS AN AUTO-GENERATED FILE. CHANGES TO IT ARE LIKELY TO BE
OVERWRITTEN AND LOST. Changes to this xml configuration should be made using:
  virsh net-edit default
or other application using the libvirt API.
-->

<network>
  <name>default</name>
  <uuid>c3459eaa-ec6e-428c-8c45-97906d71853c</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:7B:BA:7B'/>
  <ip address='192.168.120.1' netmask='255.255.248.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.127.254' />
    </dhcp>
  </ip>
</network>
[root@one-node2 ~]#
```

Nos quedamos con esta segunda forma, por ser la mas eficiente.

Ahora definimos la red de gestión (la service VLAN) de OpenNebula, por donde irá el tráfico de gestión privada del cloud, y el tráfico NFS del repositorio de imágenes.

Esto dijimos que lo haremos sobre el interfaz em3 de cada host. Usaremos sólo una clase C, pues no esperamos tener mas de 254 hosts en nuestra instalación.

Usaremos la red 10.60.1.0/24 (la IP 10.60.1.1 para el host1 y la 10.60.1.2 para el host2):

```
[root@one-node1 ~]# more /etc/sysconfig/network-scripts/ifcfg-em3
DEVICE="em3"
HWADDR="D4:AE:52:7B:AF:38"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=10.60.1.1
NETMASK=255.255.255.0
NETWORK=10.60.1.0
```

```
BROADCAST=10.60.1.255
USERCTL=no
IPV6INIT=no
PEERDNS=no
[root@one-node1 ~]# ifup em3
[root@one-node1 ~]# ifconfig em3
em3       Link encap:Ethernet  HWaddr D4:AE:52:7B:AF:38
          inet addr:10.60.1.1  Bcast:10.60.1.255  Mask:255.255.255.0
          inet6 addr: fe80::d6ae:52ff:fe7b:af38/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:408 (408.0 b)

[root@one-node1 ~]#
```

Hacemos lo mismo con el host2:

```
[root@one-node2 ~]# more /etc/sysconfig/network-scripts/ifcfg-em3
DEVICE="em3"
HWADDR="D4:AE:52:7B:AF:9F"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=10.60.1.2
NETMASK=255.255.255.0
NETWORK=10.60.1.0
BROADCAST=10.60.1.255
USERCTL=no
IPV6INIT=no
PEERDNS=no
[root@one-node2 ~]# ifup em3
[root@one-node2 ~]# ifconfig em3
em3       Link encap:Ethernet  HWaddr D4:AE:52:7B:AF:9F
          inet addr:10.60.1.2  Bcast:10.60.1.255  Mask:255.255.255.0
          inet6 addr: fe80::d6ae:52ff:fe7b:af9f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:408 (408.0 b)
```

Y probamos la conectividad entre las dos máquinas:

```
[root@one-node2 ~]# ping 10.60.1.1
PING 10.60.1.1 (10.60.1.1) 56(84) bytes of data.
64 bytes from 10.60.1.1: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 10.60.1.1: icmp_seq=2 ttl=64 time=0.308 ms
64 bytes from 10.60.1.1: icmp_seq=3 ttl=64 time=0.299 ms
```

```
64 bytes from 10.60.1.1: icmp_seq=4 ttl=64 time=0.311 ms
^C
--- 10.60.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3914ms
rtt min/avg/max/mdev = 0.299/0.526/1.188/0.382 ms
```

Por último, editamos el fichero /etc/hosts en ambas máquinas para los hosts se vean directamente a través de la nueva red de servicio montada:

```
[root@one-node1 ~]# more /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.60.1.1 one-node1 one-admin nfs-server
10.60.1.2 one-node2
[root@one-node1 ~]#

[root@one-node2 ~]# ping one-node1
PING one-node1 (10.60.1.1) 56(84) bytes of data.
64 bytes from one-node1 (10.60.1.1): icmp_seq=1 ttl=64 time=0.243 ms
64 bytes from one-node1 (10.60.1.1): icmp_seq=2 ttl=64 time=0.280 ms
64 bytes from one-node1 (10.60.1.1): icmp_seq=3 ttl=64 time=0.297 ms
^C
--- one-node1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2983ms
rtt min/avg/max/mdev = 0.243/0.273/0.297/0.026 ms
[root@one-node2 ~]# ping one-node2
PING one-node2 (10.60.1.2) 56(84) bytes of data.
64 bytes from one-node2 (10.60.1.2): icmp_seq=1 ttl=64 time=0.038 ms
64 bytes from one-node2 (10.60.1.2): icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from one-node2 (10.60.1.2): icmp_seq=3 ttl=64 time=0.025 ms
^C
--- one-node2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.025/0.030/0.038/0.007 ms
[root@one-node2 ~]#
```

2.3. Configuración del servicio NFS

Ahora que tenemos la red de gestión tenemos que compartir el repositorio de imágenes centralizado en el host1 mediante NFS para que pueda ser montado en el host2 y resto de nodos de la red. La carpeta donde se almacena el repositorio de imágenes centralizado en one-node1 es /var/lib/one/datastores

Configuramos el fichero /etc/exports en el host1.

Nota: usamos una partición física especial montada en /datastores para almacenar las imágenes del cloud:

```
[root@one-node1 one]# df -h | grep datastores
/dev/sdc1             2.0T  199M  1.9T   1% /datastores
[root@one-node1 one]#
```

A donde hemos creado un softlink en el host1, de tal forma que el directorio /var/lib/one/datatores apunte a /datastores:

```
[root@one-node1 one]# ll -d  /datastores
drwxr-x---. 5 oneadmin oneadmin 4096 Nov 14 02:08 /datastores
[root@one-node1 one]# ll /var/lib/one/
total 12
lrwxrwxrwx 1 oneadmin oneadmin   11 Nov 14 02:08 datastores -> /datastores
drwxr-x--- 4 oneadmin oneadmin 4096 Nov 13 01:00 datastores_org
drwxr-x--- 9 oneadmin oneadmin 4096 Nov 13 01:00 remotes
drwxr-x--- 2 oneadmin oneadmin 4096 Jul 24 19:13 vms
[root@one-node1 one]#

[root@one-node1 ~]# more /etc/exports
/datastores 10.60.1.0/24(rw,sync,no_subtree_check,root_squash)
[root@one-node1 ~]# exportfs -v
[root@one-node1 ~]# exportfs -a
[root@one-node1 ~]# exportfs -v
/datastores
                10.60.1.0/24(rw,wdelay,root_squash,no_subtree_check)
```

Comprobamos el estado del servicio NFS:

```
[root@one-node1 ~]# service nfs status
rpc.svcgssd is stopped
rpc.mountd is stopped
nfsd is stopped
rpc.rquotad is stopped
[root@one-node1 ~]# service nfs start
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Stopping RPC idmapd:                                       [  OK  ]
Starting RPC idmapd:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
[root@one-node1 ~]# service nfs status
rpc.svcgssd is stopped
rpc.mountd (pid 12689) is running...
nfsd (pid 12754 12753 12752 12751 12750 12749 12748 12747) is running...
rpc.rquotad (pid 12685) is running...
[root@one-node1 ~]#
```

Nos aseguramos que el servicio quedará arrancado la próxima vez que reiniciemos la máquina:

```
[root@one-node1 ~]# chkconfig nfs --list
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@one-node1 ~]# chkconfig nfs on
```

```
[root@one-node1 ~]# chkconfig nfs --list
nfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@one-node1 ~]#
```

Ahora nos aseguramos de que en el host2 el datastore queda montado y se puede acceder correctamente:

```
[oneadmin@one-node2 ~]$ ll -a
total 28
drwx------   3 oneadmin oneadmin 4096 Nov 13 01:55 .
drwxr-xr-x. 33 root     root     4096 Nov 13 01:12 ..
-rw-------   1 oneadmin oneadmin  219 Nov 14 01:41 .bash_history
-rw-r--r--   1 oneadmin oneadmin   18 Jul 18 15:19 .bash_logout
-rw-r--r--   1 oneadmin oneadmin  176 Jul 18 15:19 .bash_profile
-rw-r--r--   1 oneadmin oneadmin  124 Jul 18 15:19 .bashrc
drwx------   2 oneadmin oneadmin 4096 Nov 13 01:51 .ssh
[oneadmin@one-node2 ~]$ mkdir datastores
[oneadmin@one-node2 ~]$ chmod 750 datastores
[oneadmin@one-node2 ~]$ ll
total 4
drwxr-x--- 2 oneadmin oneadmin 4096 Nov 14 02:22 datastores
[oneadmin@one-node2 ~]$

[root@one-node2 ~]# tail -1 /etc/fstab
nfs-server:/datastores /var/lib/one/datastores  nfs   soft,intr,rsize=8192,wsize=8192,auto
[root@one-node2 ~]# mount -a
[root@one-node2 ~]# mount | grep datastores
nfs-server:/datastores on /var/lib/one/datastores type nfs
(rw,soft,intr,rsize=8192,wsize=8192,vers=4,addr=10.60.1.1,clientaddr=10.60.1.2)
[root@one-node2 ~]#
```

Bien, esta parte ya la tenemos también.

2.4. Configuración del 802.1Q para aislar las VLANs

Ahora vamos a trabajar con el interfaz em4 en los dos hosts para las redes privadas por 802.1Q de las máquinas virtuales.

Primero nos aseguramos que los interfaces físicos en los dos hosts quedarán levantados y con el STP activado aunque no configuraremos ninguna IP en la VLAN nativa (la cual está sin etiquetar), por motivos de seguridad.

```
[root@one-node1 ~]# more /etc/sysconfig/network-scripts/ifcfg-em4
DEVICE="em4"
HWADDR="D4:AE:52:7B:AF:3A"
NM_CONTROLLED="no"
ONBOOT="yes"
USERCTL=no
IPV6INIT=no
PEERDNS=no
STP="yes"
```

```
[root@one-node2 ~]# more /etc/sysconfig/network-scripts/ifcfg-em4
DEVICE="em4"
HWADDR="D4:AE:52:7B:AF:A1"
NM_CONTROLLED="no"
ONBOOT="yes"
USERCTL=no
IPV6INIT=no
PEERDNS=no
STP="yes"

[root@one-node1 ~]# ifdown em4
[root@one-node1 ~]# ifup em4
[root@one-node1 ~]# ifconfig em4
em4       Link encap:Ethernet  HWaddr D4:AE:52:7B:AF:3A
          inet6 addr: fe80::d6ae:52ff:fe7b:af3a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:250 (250.0 b)

[root@one-node2 ~]# ifdown em4
[root@one-node2 ~]# ifup em4
[root@one-node2 ~]# ifconfig em4
em4       Link encap:Ethernet  HWaddr D4:AE:52:7B:AF:A1
          inet6 addr: fe80::d6ae:52ff:fe7b:afa1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:250 (250.0 b)
```

Ahora comprobamos que carga correctamente el módulo 8021q:

```
[root@one-node1 ~]# modprobe 8021q
[root@one-node1 ~]# lsmod | grep 8021q
8021q                  25317  0
garp                    7152  1 8021q

[root@one-node2 ~]# modprobe 8021q
[root@one-node2 ~]# lsmod | grep 8021q
8021q                  25317  0
garp                    7152  1 8021q
```

Ahora tenemos que hacer que el módulo se cargue cuando arranquemos la próxima vez. Creamos el fichero /etc/sysconfig/modules/8021q.modules

```
[root@one-node1 modules]# more 8021q.modules
#!/bin/sh

modprobe -b 8021q >/dev/null 2>&1

exit 0
[root@one-node1 modules]# chmod 755 8021q.modules
[root@one-node1 modules]# ll
total 8
-rwxr-xr-x 1 root root  53 Nov 14 18:37 8021q.modules
-rwxr-xr-x 1 root root 245 Oct  2 14:54 kvm.modules
[root@one-node1 modules]#

[root@one-node2 ~]# cd  /etc/sysconfig/modules/
[root@one-node2 modules]# vi 8021q.modules
[root@one-node2 modules]# more 8021q.modules
#!/bin/sh

modprobe -b 8021q >/dev/null 2>&1

exit 0
[root@one-node2 modules]# chmod 755 8021q.modules
[root@one-node2 modules]# ll
total 8
-rwxr-xr-x 1 root root  53 Nov 14 18:40 8021q.modules
-rwxr-xr-x 1 root root 245 Oct  2 14:54 kvm.modules
[root@one-node2 modules]#
```

Además tenemos que modificar los parámetros del kernel en el fichero /etc/sysctl.conf para que las máquinas virtuales pueden verse entre si a través del interfaz bridge cuando estén conectadas a la misma red y en la misma máquina física.

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Comprobamos que ambos hosts ya lo tienen configurado OK en el fichero, pero no está configurado OK en el kernel.

```
[root@one-node1 etc]# fgrep "bridge-nf-call" sysctl.conf
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
[root@one-node1 etc]# sysctl -a | grep "bridge-nf-call"
net.bridge.bridge-nf-call-arptables = 1
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
```

Lo actualizamos ahora para que pueda funcionar, y en el siguiente arranque de la máquina los cambios serán persistentes:

```
[root@one-node1 etc]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
```

Hacemos lo mismo en el host2:

```
[root@one-node2 etc]# fgrep "bridge-nf-call" sysctl.conf
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
[root@one-node2 etc]# sysctl -a | grep "bridge-nf-call"
net.bridge.bridge-nf-call-arptables = 1
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
[root@one-node2 etc]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
```

Nota, en la maqueta comprobamos que los parámetros de desactivación de filtrado del bridge no eran persistentes entre reinicios, porque al cargar el módulo en el kernel volvían a los valores por defectos.

Para corregir ese problema tenemos que añadir una regla a las iptables que evite el filtrado del tráfico que circula dentro del bridge entre las máquinas virtuales que comparten una misma VLAN.

```
[root@one-node1 sysconfig]# diff iptables.old iptables
14a15
> -A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
[root@one-node1 sysconfig]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: mangle nat filte[  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@one-node1 sysconfig]# service iptables start
iptables: Applying firewall rules:                         [  OK  ]
[root@one-node1 sysconfig]# service libvirtd reload
Reloading libvirtd configuration:                          [  OK  ]
[root@one-node1 sysconfig]# iptables-save
Generated by iptables-save v1.4.7 on Fri Nov 15 00:02:20 2013
*mangle
:PREROUTING ACCEPT [15:1858]
:INPUT ACCEPT [15:1858]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6:744]
:POSTROUTING ACCEPT [6:744]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --checksum-fill
COMMIT
# Completed on Fri Nov 15 00:02:20 2013
# Generated by iptables-save v1.4.7 on Fri Nov 15 00:02:20 2013
*nat
:PREROUTING ACCEPT [4:998]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p tcp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p udp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -j MASQUERADE
COMMIT
# Completed on Fri Nov 15 00:02:20 2013
# Generated by iptables-save v1.4.7 on Fri Nov 15 00:02:20 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6:744]
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9869 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 29876 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.120.0/21 -o virbr0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.120.0/21 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Fri Nov 15 00:02:20 2013
```

Hacemos lo mismo con el otro host

```
[root@one-node2 sysconfig]# diff iptables iptables.old
12d11
< -A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
[root@one-node2 sysconfig]# service iptables restart
iptables: Applying firewall rules:                         [  OK  ]
[root@one-node2 sysconfig]# service libvirtd reload
Reloading libvirtd configuration:                          [  OK  ]
[root@one-node2 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Fri Nov 15 00:14:08 2013
*mangle
:PREROUTING ACCEPT [19:4579]
:INPUT ACCEPT [13:2760]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2:200]
:POSTROUTING ACCEPT [2:200]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --checksum-fill
COMMIT
# Completed on Fri Nov 15 00:14:08 2013
# Generated by iptables-save v1.4.7 on Fri Nov 15 00:14:08 2013
*nat
:PREROUTING ACCEPT [16:4327]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p tcp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p udp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -j MASQUERADE
COMMIT
# Completed on Fri Nov 15 00:14:08 2013
# Generated by iptables-save v1.4.7 on Fri Nov 15 00:14:08 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2:200]
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.120.0/21 -o virbr0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.120.0/21 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Fri Nov 15 00:14:08 2013
```

Con esto ya lo tenemos todo preparado para meternos con la configuración del OpenNebula, y arrancar los servicios.

2.5. Primeros pasos con la configuración de OpenNebula

Vamos a centrarnos en las modificaciones necesarias sobre los ficheros de configuración, en base a lo aprendido con la maqueta.

En el fichero /var/lib/one/remotes/vnm/OpenNebulaNetwork.rb tenemos que cambiar la ruta correcta para el comando brctl:

```
[oneadmin@one-admin vnm]$ diff OpenNebulaNetwork.rb OpenNebulaNetwork.rb.org
35c35
<   :brctl    => "sudo /usr/sbin/brctl",
---
>   :brctl    => "sudo /sbin/brctl",
```

Lo que nos recuerda que tenemos que dar permisos sin password en /etc/sudoers al grupo wheel para que pueda ejecutar los comandos de gestión de los interfaces de red, etc...

Descomentamos la siguiente línea con visudo en ambos hosts:

```
%wheel        ALL=(ALL)       NOPASSWD: ALL
```

Y con estos pasos, lanzamos por primera vez los servicios de opennebula:

```
[root@one-node1 ~]# service opennebula start
Starting OpenNebula daemon:                                [  OK  ]
```

```
[root@one-node1 ~]# service opennebula-sunstone start
Starting Sunstone Server daemon: VNC proxy started
sunstone-server started
                                                           [  OK  ]
[root@one-node1 ~]#
```

Vamos a ver que tal se comporta el sistema en el host1 que es el que hace también de nodo de administración:

Este es el listado de procesos lanzados:

```
oneadmin 20082     1  0 01:23 ?        00:00:00   /usr/bin/oned -f
oneadmin 20104 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_vmm_exec.rb -t 15 -r 0 
kvm
oneadmin 20116 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_im_exec.rb -r 0 -t 15 
kvm
oneadmin 20123 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_tm.rb -t 15 -d 
dummy,lvm,shared,qcow2,ssh,vmfs,iscsi,ceph
oneadmin 20133 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_hm.rb
oneadmin 20144 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_datastore.rb -t 15 -d 
dummy,fs,vmfs,iscsi,lvm,ceph
oneadmin 20154 20082  0 01:23 ?        00:00:00     ruby /usr/lib/one/mads/one_auth_mad.rb --authn 
ssh,x509,ldap,server_cipher,server_x509
oneadmin 20083     1  0 01:23 ?        00:00:00   /usr/bin/mm_sched
oneadmin 20233     1  0 01:23 ?        00:00:00   python /usr/share/one/websockify/websocketproxy.py 
--target-config=/var/lib/one/sunstone_vnc_tokens 29876
oneadmin 20239     1  0 01:23 ?        00:00:00   ruby /usr/lib/one/sunstone/sunstone-server.rb

[root@one-node1 ~]# su - oneadmin
[oneadmin@one-node1 ~]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT

[oneadmin@one-node1 ~]$
```

Bien, de momento responde bien. Veamos los logs en busca de algún problema...

Bien, todo arranca sin problemas, y no aparecen mensajes de error.

El siguiente paso es probar que podemos conectarnos a la web de administración (http://one-node1:9869) con el usuario oneadmin.

Sí, podemos entrar sin problemas. Aprovechamos para bajar algunas imágenes del Marketplace, que emplearemos después durante las pruebas de la plataforma:

```
[oneadmin@one-node1 ~]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
```

Ahora vamos a definir los dos hosts:

```
[oneadmin@one-node1 ~]$ onehost create one-node1 -i kvm -v kvm -n 802.1Q
ID: 0
[oneadmin@one-node1 ~]$ onehost create one-node2 -i kvm -v kvm -n 802.1Q
ID: 1
```

```
[oneadmin@one-node1 ~]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -           0      0 / 8000 (0%)    0K / 62.9G (0%) on
   1 one-node2       -           0      0 / 8000 (0%)    0K / 62.9G (0%) on
[oneadmin@one-node1 ~]$ onehost show 0
HOST 0 INFORMATION
ID                    : 0
NAME                  : one-node1
CLUSTER               : -
STATE                 : MONITORING_MONITORED
IM_MAD                : kvm
VM_MAD                : kvm
VN_MAD                : 802.1Q
LAST MONITORING TIME  : 11/15 01:50:51

HOST SHARES
TOTAL MEM             : 62.9G
USED MEM (REAL)       : 836.5M
USED MEM (ALLOCATED)  : 0K
TOTAL CPU             : 8000
USED CPU (REAL)       : 8
USED CPU (ALLOCATED)  : 0
RUNNING VMS           : 0

MONITORING INFORMATION
ARCH="x86_64"
CPUSPEED="2260"
FREECPU="7992.0"
FREEMEMORY="65095032"
HOSTNAME="one-node1"
HYPERVISOR="kvm"
MODELNAME="Intel(R) Xeon(R) CPU E7- 4860  @ 2.27GHz"
NETRX="0"
NETTX="0"
TOTALCPU="8000"
TOTALMEMORY="65951576"
USEDCPU="8.0"
USEDMEMORY="856544"

VIRTUAL MACHINES

    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME

[oneadmin@one-node1 ~]$ onehost show 1
HOST 1 INFORMATION
```

```
ID                     : 1
NAME                   : one-node2
CLUSTER                : -
STATE                  : MONITORED
IM_MAD                 : kvm
VM_MAD                 : kvm
VN_MAD                 : 802.1Q
LAST MONITORING TIME   : 11/15 01:50:25

HOST SHARES
TOTAL MEM              : 62.9G
USED MEM (REAL)        : 718.8M
USED MEM (ALLOCATED)   : 0K
TOTAL CPU              : 8000
USED CPU (REAL)        : 0
USED CPU (ALLOCATED)   : 0
RUNNING VMS            : 0

MONITORING INFORMATION
ARCH="x86_64"
CPUSPEED="2260"
FREECPU="8000.0"
FREEMEMORY="65215512"
HOSTNAME="one-node2"
HYPERVISOR="kvm"
MODELNAME="Intel(R) Xeon(R) CPU E7- 4860  @ 2.27GHz"
NETRX="0"
NETTX="0"
TOTALCPU="8000"
TOTALMEMORY="65951576"
USEDCPU="0.0"
USEDMEMORY="736064"

VIRTUAL MACHINES

    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME

[oneadmin@one-node1 ~]$
```

Con esto ya tenemos para empezar a definir las plantillas para las redes y los hosts de pruebas.

2.6. Definiendo las primeras platillas

Primero creamos un par de plantillas de red: una de acceso a internet por NAT, y otra de acceso a una VLAN interna.

Comenzamos por la plantilla de acceso a internet por NAT. Es la más sencilla.

Dentro del host1 (el de administración) creamos un directorio de plantillas (siempre trabajaremos con el usuario oneadmin).

```
[oneadmin@one-node1 ~]$ mkdir mytemplates
[oneadmin@one-node1 ~]$ cd mytemplates/
[oneadmin@one-node1 mytemplates]$ more public_nat.net
NAME = "Internet NAT LAN"
TYPE = "RANGED"
BRIDGE = "virbr0"
VLAN = NO

NETWORK_ADDRESS = "192.168.120.0/21"
GATEWAY = "192.168.120.1"
DNS = "192.168.120.1"
IP_START = "192.168.120.2"
IP_END = "192.168.127.254"

[oneadmin@one-node1 mytemplates]$ onevnet create public_nat.net
ID: 0
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   0 oneadmin     oneadmin     Internet NAT LA -               R virbr0        0
[oneadmin@one-node1 mytemplates]$ onevnet show 0
VIRTUAL NETWORK 0 INFORMATION
ID             : 0
NAME           : Internet NAT LAN
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : virbr0
VLAN           : No
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="192.168.120.1"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
```

```
IP_START       : 192.168.120.2
IP_END         : 192.168.127.254

VIRTUAL MACHINES

```

Hacemos que esta plantilla pueda utilizarla cualquier usuario, y no sólo oneadmin, que es el administrador.

```
[oneadmin@one-node1 mytemplates]$ onevnet chmod 0 644
[oneadmin@one-node1 mytemplates]$ onevnet show 0
VIRTUAL NETWORK 0 INFORMATION
ID             : 0
NAME           : Internet NAT LAN
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : virbr0
VLAN           : No
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

VIRTUAL NETWORK TEMPLATE
DNS="192.168.120.1"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.2
IP_END         : 192.168.127.254

VIRTUAL MACHINES

```

Ahora creamos otra plantilla para la VLAN 2 (no debemos usar nunca la VLAN nativa, por motivos de seguridad):

```
[oneadmin@one-node1 mytemplates]$ more priv_vlan2.net
NAME = "VLAN2"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 2
```

```
BRIDGE  = "brvl2"

NETWORK_ADDRESS = "10.132.120.0/21"

[oneadmin@one-node1 mytemplates]$ onevnet create priv_vlan2.net
ID: 1
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   0 oneadmin     oneadmin     Internet NAT LA -               R virbr0        0
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
[oneadmin@one-node1 mytemplates]$ onevnet show 1
VIRTUAL NETWORK 1 INFORMATION
ID             : 1
NAME           : VLAN2
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl2
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 2
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="10.132.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 10.132.120.1
IP_END         : 10.132.127.254

VIRTUAL MACHINES
```

Bien, ahora creamos una plantilla para poder lanzar una máquina virtual de pruebas. Definiremos dos interfaces de red, usando las dos plantillas definidas anteriormente:

```
[oneadmin@one-node1 mytemplates]$ more tty_2nic_1.host
NAME="tty test 2 NICs 1"
CPU="0.1"
DISK=[
  IMAGE_ID="0" ]
```

```
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
FEATURES=[
  ACPI="no" ]
NIC = [ NETWORK="Internet NAT LAN" ]
NIC = [ NETWORK="VLAN2" ]

[oneadmin@one-node1 mytemplates]$ onetemplate create tty_2nic_1.host
ID: 0
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER            GROUP           NAME                                REGTIME
   0 oneadmin        oneadmin        tty test 2 NICs 1            11/16 01:29:12
[oneadmin@one-node1 mytemplates]$ onetemplate show 0
TEMPLATE 0 INFORMATION
ID             : 0
NAME           : tty test 2 NICs 1
USER           : oneadmin
GROUP          : oneadmin
REGISTER TIME  : 11/16 01:29:12

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CPU="0.1"
DISK=[
  IMAGE_ID="0" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="Internet NAT LAN" ]
NIC=[
  NETWORK="VLAN2" ]
[oneadmin@one-node1 mytemplates]$
```

Con esto ya tenemos preparado todo para instanciar nuestra primera máquina virtual. Vamos a probarlo:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 0 --name "tty_public1"
VM ID: 0
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     fail    0      0K               0d 00h00
[oneadmin@one-node1 mytemplates]$
```

Era de esperar. Vamos a ver lo que ocurre en los logs:

```
Sat Nov 16 01:33:02 2013 [VMM][I]: sudo: sorry, you must have a tty to run sudo
```

Se nos había pasado por alto el problema que había con el sudo en CentOS/RedHat/Fedora y tiene que ver con la siguientes líneas en el fichero /etc/sudoers:

```
Defaults requiretty
Defaults !visiblepw
```

Comentando esas línea en el /etc/sudoers de los hosts el tema del sudo queda resuelto. Lanzamos de nuevo el comando:

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     fail    0      0K               0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm delete 0 --recreate
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     pend    0      0K               0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    1     64M one-node2     0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm show 0
VIRTUAL MACHINE 0 INFORMATION
ID                  : 0
NAME                : tty_public1
USER                : oneadmin
GROUP               : oneadmin
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node2
START TIME          : 11/16 01:32:52
END TIME            : 11/16 01:33:02
DEPLOY ID           : one-0

VIRTUAL MACHINE MONITORING
NET_RX              : 1K
USED CPU            : 1
NET_TX              : 0K
USED MEMORY         : 64M
```

```
PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda     ttylinux                           file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet NAT LAN       no virbr0       192.168.120.2   02:00:c0:a8:78:02
                                           fe80::400:c0ff:fea8:7802
  1 VLAN2                 yes brvl2        10.132.120.1    02:00:0a:84:78:01
                                           fe80::400:aff:fe84:7801

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node2       none             erro  11/16 01:33:01   0d 00h00m   0h00m01s
  1 one-node2       none             none  11/16 01:45:01   0d 00h00m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CPU="0.1"
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="5900",
  TYPE="VNC" ]
MEMORY="64"
TEMPLATE_ID="0"
VMID="0"
[oneadmin@one-node1 mytemplates]$
```

Ahora sí ha funcionado. Se encuentra funcionando en el host2 (one-node2).

```
[oneadmin@one-node1 mytemplates]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -           0       0 / 8000 (0%)     0K / 62.9G (0%) on
   1 one-node2       -           1      10 / 8000 (0%)    64M / 62.9G (0%) on
[oneadmin@one-node1 mytemplates]$
```

Con los recursos asignados, podríamos lanzar cerca de 800 máquinas virtuales en cada host (tenemos 80 CPUs por host, y hemos especificado en la plantilla que queríamos un 10% de CPU). Comprobamos que los recursos de red han quedado bien definidos en el host2:

```
[root@one-node2 ~]# brctl show
bridge name     bridge id               STP enabled     interfaces
brvl2           8000.d4ae527bafa1       no              em4.2
                                                        vnet1
virbr0          8000.5254007bba7b       yes             virbr0-nic
                                                        vnet0
[root@one-node2 ~]# ping 192.168.120.2
PING 192.168.120.2 (192.168.120.2) 56(84) bytes of data.
64 bytes from 192.168.120.2: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from 192.168.120.2: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 192.168.120.2: icmp_seq=3 ttl=64 time=0.549 ms
64 bytes from 192.168.120.2: icmp_seq=4 ttl=64 time=0.226 ms
64 bytes from 192.168.120.2: icmp_seq=5 ttl=64 time=0.288 ms
^C
--- 192.168.120.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4683ms
rtt min/avg/max/mdev = 0.226/0.562/1.484/0.474 ms
[root@one-node2 ~]# ssh root@192.168.120.2
The authenticity of host '192.168.120.2 (192.168.120.2)' can't be established.
RSA key fingerprint is 5b:d6:3a:a9:8a:53:21:66:70:0c:b7:26:34:45:b1:27.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.120.2' (RSA) to the list of known hosts.
root@192.168.120.2's password:

Chop wood, carry water.

# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:C0:A8:78:02
          inet addr:192.168.120.2  Bcast:192.168.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5231 (5.1 KiB)  TX bytes:4015 (3.9 KiB)
          Interrupt:11 Base address:0xc100

eth1      Link encap:Ethernet  HWaddr 02:00:0A:84:78:01
          inet addr:10.132.120.1  Bcast:10.132.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:11 Base address:0xc200
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                38.6M      8.0M     28.6M  22% /
tmpfs                    24.0K         0     24.0K   0% /dev/shm
# free
              total         used         free       shared      buffers
  Mem:        60244         8332        51912            0          216
 Swap:            0            0            0
Total:        60244         8332        51912
# ping 192.100.0.250
PING 192.100.0.250 (192.100.0.250): 56 data bytes
64 bytes from 192.100.0.250: seq=0 ttl=253 time=1.296 ms
64 bytes from 192.100.0.250: seq=1 ttl=253 time=1.413 ms
64 bytes from 192.100.0.250: seq=2 ttl=253 time=1.129 ms
64 bytes from 192.100.0.250: seq=3 ttl=253 time=1.054 ms
64 bytes from 192.100.0.250: seq=4 ttl=253 time=1.082 ms
64 bytes from 192.100.0.250: seq=5 ttl=253 time=1.145 ms

--- 192.100.0.250 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 1.054/1.186/1.413 ms

# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.120.0   0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.132.120.0    0.0.0.0         255.255.255.0   U         0 0          0 eth1
0.0.0.0         192.168.120.1   0.0.0.0         UG        0 0          0 eth0
#
```

Bien, vamos a instanciar otra máquina del mismo tipo, para confirmar que la red por 802.1Q funciona entre hosts, y que las máquinas se ven entre si por la red privada:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 0 --name "tty_public2"
VM ID: 1
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    0      0K one-node1    0d 00h00
```

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    0      0K one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    0      0K one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn   94     64M one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn   94     64M one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn   94     64M one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm show 1
VIRTUAL MACHINE 1 INFORMATION
ID                    : 1
NAME                  : tty_public2
USER                  : oneadmin
GROUP                 : oneadmin
STATE                 : ACTIVE
LCM_STATE             : RUNNING
RESCHED               : No
HOST                  : one-node1
START TIME            : 11/16 02:03:57
END TIME              : -
DEPLOY ID             : one-1

VIRTUAL MACHINE MONITORING
NET_TX                : 0K
USED CPU              : 94
USED MEMORY           : 64M
NET_RX                : 1K

PERMISSIONS
OWNER                 : um-
GROUP                 : ---
OTHER                 : ---
```

```
VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    ttylinux                            file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet NAT LAN       no virbr0       192.168.120.3   02:00:c0:a8:78:03
                                           fe80::400:c0ff:fea8:7803
  1 VLAN2                 yes brvl2        10.132.120.2    02:00:0a:84:78:02
                                           fe80::400:aff:fe84:7802

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node1       none             none  11/16 02:04:01    0d 00h00m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CPU="0.1"
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="5901",
  TYPE="VNC" ]
MEMORY="64"
TEMPLATE_ID="0"
VMID="1"
```

Vamos a probarlo:

```
[root@one-node1 ~]# ping 192.168.120.3
PING 192.168.120.3 (192.168.120.3) 56(84) bytes of data.
64 bytes from 192.168.120.3: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 192.168.120.3: icmp_seq=2 ttl=64 time=0.465 ms
64 bytes from 192.168.120.3: icmp_seq=3 ttl=64 time=0.223 ms
^C
--- 192.168.120.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2734ms
rtt min/avg/max/mdev = 0.223/0.680/1.354/0.487 ms
[root@one-node1 ~]# brctl show
bridge name     bridge id               STP enabled     interfaces
brvl2           8000.d4ae527baf3a       no              em4.2
                                                        vnet1
virbr0          8000.525400917e04       yes             virbr0-nic
                                                        vnet0
[root@one-node1 ~]# ssh root@192.168.120.3
The authenticity of host '192.168.120.3 (192.168.120.3)' can't be established.
```

```
RSA key fingerprint is 5b:d6:3a:a9:8a:53:21:66:70:0c:b7:26:34:45:b1:27.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.120.3' (RSA) to the list of known hosts.
root@192.168.120.3's password:

Chop wood, carry water.

# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:C0:A8:78:03
          inet addr:192.168.120.3  Bcast:192.168.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4885 (4.7 KiB)  TX bytes:3845 (3.7 KiB)
          Interrupt:11 Base address:0xc100

eth1      Link encap:Ethernet  HWaddr 02:00:0A:84:78:02
          inet addr:10.132.120.2  Bcast:10.132.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:11 Base address:0xc200

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

# ping 192.100.0.250
PING 192.100.0.250 (192.100.0.250): 56 data bytes
64 bytes from 192.100.0.250: seq=0 ttl=253 time=23.934 ms
64 bytes from 192.100.0.250: seq=1 ttl=253 time=25.567 ms

--- 192.100.0.250 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 23.934/24.750/25.567 ms
```

```
# ping 10.132.120.1
PING 10.132.120.1 (10.132.120.1): 56 data bytes
64 bytes from 10.132.120.1: seq=0 ttl=64 time=1.991 ms
64 bytes from 10.132.120.1: seq=1 ttl=64 time=0.865 ms
64 bytes from 10.132.120.1: seq=2 ttl=64 time=0.609 ms
64 bytes from 10.132.120.1: seq=3 ttl=64 time=0.603 ms

--- 10.132.120.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.603/1.017/1.991 ms
```

Funciona: tenemos tanto acceso al exterior como desde dentro de la VLAN2 entre las dos máquinas virtuales, cada una de ellas corriendo en un host diferente:

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    5     64M one-node1    0d 00h09
[oneadmin@one-node1 mytemplates]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -           1     10 / 8000 (0%)   64M / 62.9G (0%) on
   1 one-node2       -           1     10 / 8000 (0%)   64M / 62.9G (0%) on
```

2.7. Realizando pruebas de migración en vivo de las instancias

Ahora vamos a probar a hacer una migración en vivo de una de las máquinas (la que acabamos de crear, por ejemplo):

```
[oneadmin@one-node1 mytemplates]$ onevm migrate 1 one-node2 --live -v
VM 1: migrating to 1
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node2    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    5     64M one-node2    0d 00h15
```

Ahora probamos a hacer justo lo contrario: pasaremos la vm tty_public1 al host1.

```
[oneadmin@one-node1 mytemplates]$ onevm migrate 0 one-node1 --live -v
VM 0: migrating to 0
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    6     64M one-node1    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    5     64M one-node2    0d 00h20
```

Ahora migramos la vm tty_public2 al host1, manteniendo un ping continuo al interfaz vlan desde la vm tty_public1:

```
[root@one-node1 ~]# ssh root@192.168.120.2
The authenticity of host '192.168.120.2 (192.168.120.2)' can't be established.
RSA key fingerprint is 5b:d6:3a:a9:8a:53:21:66:70:0c:b7:26:34:45:b1:27.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.120.2' (RSA) to the list of known hosts.
root@192.168.120.2's password:
```

```
Chop wood, carry water.

# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:C0:A8:78:02
          inet addr:192.168.120.2  Bcast:192.168.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:577 errors:0 dropped:0 overruns:0 frame:0
          TX packets:379 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:54595 (53.3 KiB)  TX bytes:54797 (53.5 KiB)
          Interrupt:11 Base address:0xc100

eth1      Link encap:Ethernet  HWaddr 02:00:0A:84:78:01
          inet addr:10.132.120.1  Bcast:10.132.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:355 errors:0 dropped:0 overruns:0 frame:0
          TX packets:350 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33536 (32.7 KiB)  TX bytes:33236 (32.4 KiB)
          Interrupt:11 Base address:0xc200

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1608 (1.5 KiB)  TX bytes:1608 (1.5 KiB)

# ping 10.132.120.2
PING 10.132.120.2 (10.132.120.2): 56 data bytes
64 bytes from 10.132.120.2: seq=0 ttl=64 time=0.769 ms
64 bytes from 10.132.120.2: seq=1 ttl=64 time=0.754 ms
64 bytes from 10.132.120.2: seq=2 ttl=64 time=0.684 ms
64 bytes from 10.132.120.2: seq=3 ttl=64 time=0.595 ms
64 bytes from 10.132.120.2: seq=4 ttl=64 time=0.462 ms
64 bytes from 10.132.120.2: seq=5 ttl=64 time=0.558 ms
64 bytes from 10.132.120.2: seq=6 ttl=64 time=0.564 ms
64 bytes from 10.132.120.2: seq=7 ttl=64 time=0.562 ms
64 bytes from 10.132.120.2: seq=8 ttl=64 time=0.589 ms
64 bytes from 10.132.120.2: seq=9 ttl=64 time=0.613 ms
64 bytes from 10.132.120.2: seq=10 ttl=64 time=0.617 ms
64 bytes from 10.132.120.2: seq=11 ttl=64 time=0.685 ms
64 bytes from 10.132.120.2: seq=12 ttl=64 time=0.654 ms
```

```
64 bytes from 10.132.120.2: seq=13 ttl=64 time=0.563 ms
64 bytes from 10.132.120.2: seq=14 ttl=64 time=0.577 ms
64 bytes from 10.132.120.2: seq=15 ttl=64 time=0.626 ms
64 bytes from 10.132.120.2: seq=16 ttl=64 time=0.617 ms
64 bytes from 10.132.120.2: seq=17 ttl=64 time=0.589 ms
64 bytes from 10.132.120.2: seq=18 ttl=64 time=0.612 ms
64 bytes from 10.132.120.2: seq=19 ttl=64 time=0.569 ms
64 bytes from 10.132.120.2: seq=20 ttl=64 time=0.673 ms
64 bytes from 10.132.120.2: seq=21 ttl=64 time=0.607 ms
64 bytes from 10.132.120.2: seq=22 ttl=64 time=0.678 ms
64 bytes from 10.132.120.2: seq=23 ttl=64 time=0.656 ms
64 bytes from 10.132.120.2: seq=24 ttl=64 time=0.595 ms
64 bytes from 10.132.120.2: seq=25 ttl=64 time=0.632 ms
64 bytes from 10.132.120.2: seq=26 ttl=64 time=0.705 ms
64 bytes from 10.132.120.2: seq=27 ttl=64 time=0.517 ms
64 bytes from 10.132.120.2: seq=28 ttl=64 time=0.685 ms
64 bytes from 10.132.120.2: seq=29 ttl=64 time=0.682 ms
64 bytes from 10.132.120.2: seq=30 ttl=64 time=0.551 ms
64 bytes from 10.132.120.2: seq=31 ttl=64 time=0.711 ms
64 bytes from 10.132.120.2: seq=32 ttl=64 time=0.508 ms
64 bytes from 10.132.120.2: seq=33 ttl=64 time=0.509 ms
64 bytes from 10.132.120.2: seq=34 ttl=64 time=0.480 ms
64 bytes from 10.132.120.2: seq=35 ttl=64 time=0.470 ms
64 bytes from 10.132.120.2: seq=36 ttl=64 time=0.462 ms
64 bytes from 10.132.120.2: seq=37 ttl=64 time=0.514 ms
64 bytes from 10.132.120.2: seq=38 ttl=64 time=0.612 ms
64 bytes from 10.132.120.2: seq=39 ttl=64 time=0.553 ms
64 bytes from 10.132.120.2: seq=40 ttl=64 time=0.642 ms
64 bytes from 10.132.120.2: seq=41 ttl=64 time=0.554 ms
64 bytes from 10.132.120.2: seq=42 ttl=64 time=0.593 ms
64 bytes from 10.132.120.2: seq=43 ttl=64 time=0.560 ms
64 bytes from 10.132.120.2: seq=44 ttl=64 time=0.541 ms
64 bytes from 10.132.120.2: seq=45 ttl=64 time=0.715 ms
64 bytes from 10.132.120.2: seq=46 ttl=64 time=0.669 ms
64 bytes from 10.132.120.2: seq=47 ttl=64 time=0.637 ms
64 bytes from 10.132.120.2: seq=48 ttl=64 time=0.360 ms
64 bytes from 10.132.120.2: seq=49 ttl=64 time=0.303 ms
64 bytes from 10.132.120.2: seq=50 ttl=64 time=0.298 ms
64 bytes from 10.132.120.2: seq=51 ttl=64 time=0.279 ms
64 bytes from 10.132.120.2: seq=52 ttl=64 time=0.363 ms
64 bytes from 10.132.120.2: seq=53 ttl=64 time=0.504 ms
64 bytes from 10.132.120.2: seq=54 ttl=64 time=0.359 ms
64 bytes from 10.132.120.2: seq=55 ttl=64 time=0.385 ms
64 bytes from 10.132.120.2: seq=56 ttl=64 time=0.464 ms
64 bytes from 10.132.120.2: seq=57 ttl=64 time=0.347 ms
```

```
64 bytes from 10.132.120.2: seq=58 ttl=64 time=0.771 ms
64 bytes from 10.132.120.2: seq=59 ttl=64 time=0.600 ms
64 bytes from 10.132.120.2: seq=60 ttl=64 time=0.334 ms
64 bytes from 10.132.120.2: seq=61 ttl=64 time=0.282 ms
64 bytes from 10.132.120.2: seq=62 ttl=64 time=0.384 ms
64 bytes from 10.132.120.2: seq=63 ttl=64 time=0.314 ms
64 bytes from 10.132.120.2: seq=64 ttl=64 time=0.605 ms
64 bytes from 10.132.120.2: seq=65 ttl=64 time=0.659 ms
64 bytes from 10.132.120.2: seq=66 ttl=64 time=0.638 ms
64 bytes from 10.132.120.2: seq=67 ttl=64 time=0.281 ms

--- 10.132.120.2 ping statistics ---
68 packets transmitted, 68 packets received, 0% packet loss
round-trip min/avg/max = 0.279/0.551/0.771 ms
```

Nada: no se ha perdido ni un solo paquete durante la migración. Aquí podemos ver cómo lo hemos migrado:

```
[oneadmin@one-node1 mytemplates]$ onevm migrate 1 one-node1 --live -v
VM 1: migrating to 0
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
     0 oneadmin oneadmin tty_public1     runn    5     64M one-node1    0d 00h00
     1 oneadmin oneadmin tty_public2     runn    5     64M one-node1    0d 00h26
[oneadmin@one-node1 mytemplates]$
```

Sencillamente: es increíble.

2.8. Optimizando la instanciación de las imágenes en los DS

Vamos a comprobar la velocidad del disco duro para poder instanciar una imagen de 11G (Ubuntu 12.04). Copiamos la imagen desde el datastore a otro fichero desde el mismo directorio y medimos el tiempo:

```
[oneadmin@one-node1 1]$ ll
total 33800208
-rw-r--r-- 1 oneadmin oneadmin 11811160064 Nov 15 02:05 4d3f33bdb1147e07b5322d6920683a15
-rw-r--r-- 1 oneadmin oneadmin 11811160064 Nov 16 00:37 644079a4862479b8e0c26279acd1f058
-rw-r--r-- 1 oneadmin oneadmin 10737418240 Nov 16 01:07 9a11234cab1c05898e22b048b157cd83
-rw-r--r-- 1 oneadmin oneadmin    41943040 Nov 15 01:44 d64639d03817c24dd97b35208507f511
-rw-r--r-- 1 oneadmin oneadmin   209715201 Nov 15 01:47 e83b66bcb88a4f7be1cd41c9d16bb314
[oneadmin@one-node1 1]$ time cp 4d3f33bdb1147e07b5322d6920683a15
4d3f33bdb1147e07b5322d6920683a15.copy

real    0m18.758s
user    0m0.061s
sys     0m18.588s
[oneadmin@one-node1 1]$
```

Ya sabemos, que en el mejor de los casos, instanciar la imagen de Ubuntu desde el mismo host (one-node1) donde se encuentra el repositorio, tardaría como mínimo 19 segundos en estar preparado para arrancar la imagen.

Tenemos que ajustar el TM para evitar que la copia del fichero utilice el interfaz de red cuando el host donde correrá la instancia sea el host2.

Podemos ver la diferencia haciendo la misma operación sobre el host1 y sobre el host2 (el cual tiene montada el datastore por NFS).

```
[oneadmin@one-node1 1]$ time cp d64639d03817c24dd97b35208507f511
d64639d03817c24dd97b35208507f511.copy


real    0m0.053s
user    0m0.001s
sys     0m0.052s
[oneadmin@one-node1 1]$


[oneadmin@one-node2 1]$ time cp d64639d03817c24dd97b35208507f511
d64639d03817c24dd97b35208507f511.copy


real    0m0.458s
user    0m0.000s
sys     0m0.081s
[oneadmin@one-node2 1]$
```

La diferencia entre hacer la copia local y a través de la red es notable, incluso con la imagen de 40M.

El truco para acelerar la velocidad de clonación de la imagen se explica aquí:

```
http://wiki.opennebula.org/local_scp_tm_shared
http://lists.opennebula.org/pipermail/users-opennebula.org/2012-April/008531.html
```

Modificamos el script /var/lib/one/remotes/tm/shared/clone con lo siguiente:

```
[oneadmin@one-node1 shared]$ diff clone clone.org
45,46c45
< #DST_HOST=`arg_host $DST`
< DST_HOST="one-node1"
---
> DST_HOST=`arg_host $DST`
```

Hacemos un onehost sync para que el script se actualice en el host 2 (en los remotes):

```
[oneadmin@one-node1 ~]$ onehost sync
[oneadmin@one-node1 ~]$
```

En el siguiente ciclo de control, el host 2 quedará actualizado con el cambio de script. Comprobamos que se ha actualizado el contenido del directorio /var/tmp/one/tm/shared en el host2:

```
[oneadmin@one-node2 shared]$ pwd
/var/tmp/one/tm/shared
```

```
[oneadmin@one-node2 shared]$ ll
total 48
-rwxr-xr-x 1 oneadmin oneadmin 3059 Nov 16 03:53 clone
-rwxr-xr-x 1 oneadmin oneadmin 3037 Nov 16 03:53 clone.org
-rwxr-xr-x 1 oneadmin oneadmin 3844 Nov 16 03:53 context
-rwxr-xr-x 1 oneadmin oneadmin 2839 Nov 16 03:53 cpds
-rwxr-xr-x 1 oneadmin oneadmin 2109 Nov 16 03:53 delete
-rwxr-xr-x 1 oneadmin oneadmin 2817 Nov 16 03:53 ln
-rwxr-xr-x 1 oneadmin oneadmin 2582 Nov 16 03:53 mkimage
-rwxr-xr-x 1 oneadmin oneadmin 1596 Nov 16 03:53 mkswap
-rwxr-xr-x 1 oneadmin oneadmin 1204 Nov 16 03:53 mv
-rwxr-xr-x 1 oneadmin oneadmin 2854 Nov 16 03:53 mvds
-rwxr-xr-x 1 oneadmin oneadmin 1682 Nov 16 03:53 postmigrate
-rwxr-xr-x 1 oneadmin oneadmin 1681 Nov 16 03:53 premigrate
[oneadmin@one-node2 shared]$
```

Vemos que se ha copiado todo el directorio. Vamos a migrar los servidores del direccionamiento privado privado actual, a direccionamiento público.

3. Migración de los host a direccionamiento público

Lo primero es parar todas las máquinas virtuales que tenemos en funcionamiento, ya que están haciendo NAT a través del interfaz bridge que usa KVM por defecto:

```
[oneadmin@one-node1 ~]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -         105  1050 / 8000 (13%) 6.6G / 62.9G (10%) update
   1 one-node2       -         107  1070 / 8000 (13%) 6.7G / 62.9G (10%) update
[oneadmin@one-node1 ~]$ onevm delete 0..211
[oneadmin@one-node1 ~]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
```

Ahora podemos trabajar con el tema de desactivar la red "default" utilizada por libvirt para KVM.

```
[root@one-node1 log]# virsh net-list
Name                 State      Autostart     Persistent
--------------------------------------------------
default              active     yes           yes

[root@one-node1 log]# virsh net-info default
Name            default
UUID            c93540ed-c9fa-4079-a4c7-2e75ca0ad673
Active:         yes
Persistent:     yes
Autostart:      yes
Bridge:         virbr0
```

```
[root@one-node1 log]# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>c93540ed-c9fa-4079-a4c7-2e75ca0ad673</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:91:7E:04'/>
  <ip address='192.168.120.1' netmask='255.255.248.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.127.254' />
    </dhcp>
  </ip>
</network>

[root@one-node1 log]# virsh net-destroy default
Network default destroyed

[root@one-node1 log]# virsh net-info default
Name            default
UUID            c93540ed-c9fa-4079-a4c7-2e75ca0ad673
Active:         no
Persistent:     yes
Autostart:      yes
Bridge:         virbr0

[root@one-node1 log]# virsh net-autostart default  --disable
Network default unmarked as autostarted

[root@one-node1 log]# virsh net-info default
Name            default
UUID            c93540ed-c9fa-4079-a4c7-2e75ca0ad673
Active:         no
Persistent:     yes
Autostart:      no
Bridge:         virbr0
```

Con esto nos aseguramos que la próxima vez que se reinicie la máquina no se quede activado, y el tráfico de las máquinas virtuales pase a través del mismo interfaz utilizado para la gestión del servidor. A partir de ahora, todo el tráfico de las máquinas virtuales pasará a través de los routers virtuales, o bien a través de una IP pública dedicada, pero saldrá a través de un interfaz de red físico diferente del interfaz de gestión del host.

Después de desactivar la red en el host1, comprobamos que las reglas de NAT aplicadas al interfaz virbr0 han desaparecido:

```
[root@one-node1 log]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:02:13 2013
```

```
*mangle
:PREROUTING ACCEPT [1739:344172]
:INPUT ACCEPT [1547:273966]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [991:157030]
:POSTROUTING ACCEPT [991:157030]
COMMIT
# Completed on Tue Dec 17 16:02:13 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:02:13 2013
*nat
:PREROUTING ACCEPT [655:182834]
:POSTROUTING ACCEPT [21:1292]
:OUTPUT ACCEPT [21:1292]
COMMIT
# Completed on Tue Dec 17 16:02:13 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:02:13 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [991:157030]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9869 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 29876 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Dec 17 16:02:13 2013
```

Para que se vea la diferencia con la situación anterior, vamos a hacer lo mismo con el otro host:

```
[root@one-node2 ~]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:11:37 2013
*mangle
:PREROUTING ACCEPT [11724715:3337868102]
:INPUT ACCEPT [10136148:2748494174]
:FORWARD ACCEPT [42:3528]
:OUTPUT ACCEPT [6962359:1759641074]
:POSTROUTING ACCEPT [6962401:1759644602]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --checksum-fill
COMMIT
```

```
# Completed on Tue Dec 17 16:11:37 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:11:37 2013
*nat
:PREROUTING ACCEPT [5472892:1483654552]
:POSTROUTING ACCEPT [8264:628028]
:OUTPUT ACCEPT [8264:628028]
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p tcp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -p udp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.120.0/21 ! -d 192.168.120.0/21 -j MASQUERADE
COMMIT
# Completed on Tue Dec 17 16:11:37 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:11:37 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6962360:1759642054]
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.120.0/21 -o virbr0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.120.0/21 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Dec 17 16:11:37 2013
```

Desactivamos la red default:

```
[root@one-node2 ~]# virsh net-list
Name                 State      Autostart     Persistent
--------------------------------------------------
default              active     yes           yes

[root@one-node2 ~]# virsh net-info default
Name            default
UUID            c3459eaa-ec6e-428c-8c45-97906d71853c
```

```
Active:         yes
Persistent:     yes
Autostart:      yes
Bridge:         virbr0

[root@one-node2 ~]# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>c3459eaa-ec6e-428c-8c45-97906d71853c</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:7B:BA:7B'/>
  <ip address='192.168.120.1' netmask='255.255.248.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.127.254' />
    </dhcp>
  </ip>
</network>

[root@one-node2 ~]# virsh net-destroy default
Network default destroyed

[root@one-node2 ~]# virsh net-info default
Name            default
UUID            c3459eaa-ec6e-428c-8c45-97906d71853c
Active:         no
Persistent:     yes
Autostart:      yes
Bridge:         virbr0

[root@one-node2 ~]# virsh net-autostart default  --disable
Network default unmarked as autostarted

[root@one-node2 ~]# virsh net-info default
Name            default
UUID            c3459eaa-ec6e-428c-8c45-97906d71853c
Active:         no
Persistent:     yes
Autostart:      no
Bridge:         virbr0

[root@one-node2 ~]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:14:19 2013
*mangle
:PREROUTING ACCEPT [72:12522]
```

```
:INPUT ACCEPT [62:9184]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [29:3608]
:POSTROUTING ACCEPT [29:3608]
COMMIT
# Completed on Tue Dec 17 16:14:19 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:14:19 2013
*nat
:PREROUTING ACCEPT [37:10110]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
# Completed on Tue Dec 17 16:14:19 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:14:19 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [29:3608]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Dec 17 16:14:19 2013
```

Como parte del bastionado, migraremos el puerto ssh al 2222, y descartaremos silenciosamente el resto de paquetes que nos lleguen. De esta forma, solo los puertos del interfaz de gestión quedarán accesibles desde fuera, y como son puertos altos pasarán desapercibidos a los scaneos de IPs atacantes. El interfaz físico con acceso al exterior será el em1, y será el único que tenga una IP pública. Vamos a editar el fichero de iptables. Primero trabajaremos en el host 2, que es un servidor puro de virtualización.

```
[root@one-node2 ~]# cd /etc/sysconfig/
[root@one-node2 sysconfig]# more iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Y lo dejamos así:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2222 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
```

Vamos a aplicar la reglas en el host2:

```
[root@one-node2 sysconfig]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: mangle nat filte[  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@one-node2 sysconfig]# service iptables start
iptables: Applying firewall rules:                         [  OK  ]
[root@one-node2 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 16:47:35 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [296:52108]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2222 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
# Completed on Tue Dec 17 16:47:35 2013
```

Vamos a permitir que el proceso sshd pueda escuchar en el puerto 22 y el 2222 a un mismo tiempo, ya que todos los procesos de gestión de OpenNebula se apoyan en operaciones mediante ssh y con eso evitamos modificar todos los ficheros de configuración para que utilicen el puerto 2222. En el fichero /etc/ssh/sshd_config descomentamos la linea que define el puerto 22 por defecto y añadimos una nueva linea para que escuche también al puerto 2222:

```
Port 22
Port 2222
```

Recargamos la configuración del proceso:

```
[root@one-node2 ssh]# service sshd reload
Reloading sshd:                                            [  OK  ]
```

Comprobamos que se está escuchando en ambos puertos, aunque como seguimos con la sesión ssh establecida, sabemos que de momento todo va bien:

```
[root@one-node2 ssh]# nmap -sT localhost

Starting Nmap 5.51 ( http://nmap.org ) at 2013-12-17 17:03 CET
Failed to find device em4 which was referenced in /proc/net/route
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00034s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
111/tcp  open  rpcbind
631/tcp  open  ipp
2222/tcp open  EtherNet/IP-1

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

Ahora, si lo hacemos desde otra máquina de fuera de la red, veremos lo siguiente:

```
[jamontes@Tesla ~]$ ping one-node2
PING one-node2 (192.168.25.204) 56(84) bytes of data.
^C
--- one-node2 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 4999ms

[jamontes@Tesla ~]$ nmap -sT one-node2

Starting Nmap 6.01 ( http://nmap.org ) at 2013-12-17 17:15 CET
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.04 seconds
[jamontes@Tesla ~]$ ssh root@one-node2 -p 2222
Last login: Tue Dec 17 13:38:56 2013 from 10.134.16.105
[root@one-node2 ~]#  exit
```

```
logout
Connection to one-node2 closed.
[jamontes@Tesla ~]$ ssh root@one-node2
Last login: Tue Dec 17 17:16:07 2013 from 10.134.16.105
[root@one-node2 ~]#
```

Vemos que el tema ha funcionado OK. El servidor escucha en ambos puertos el servicio ssh. El siguiente paso será permitir por iptables sólo el puerto 2222, dejando el 22 para uso de la red de gestión interna. De esta forma el servidor será prácticamente invisible desde fuera. Ahora nos queda cambiar el direccionamiento del puerto de salida a internet, para que sea direccionamiento 100% público. Entramos al equipo desde la red privada a través del host1:

```
[root@one-node1 ~]# ssh one-node2
Warning: Permanently added the RSA host key for IP address '10.60.1.2' to the list of known hosts.
root@one-node2's password:
Last login: Tue Dec 17 17:17:28 2013 from 10.134.16.105
[root@one-node2 ~]# who
root     pts/0        2013-12-17 17:22 (10.60.1.1)
[root@one-node2 ~]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.60.1.0       0.0.0.0         255.255.255.0   U         0 0          0 em3
192.168.25.0    0.0.0.0         255.255.255.0   U         0 0          0 em1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em4
0.0.0.0         192.168.25.254  0.0.0.0         UG        0 0          0 em1
```

Vemos que la salida al exterior se realiza a través del interfaz em1, que será el que migremos. Tenemos esto:

```
[root@one-node2 ~]# more /etc/sysconfig/network-scripts/ifcfg-em1
DEVICE="em1"
HWADDR="D4:AE:52:7B:AF:9B"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=192.168.25.204
NETMASK=255.255.255.0
NETWORK=192.168.25.0
BROADCAST=192.168.25.255
USERCTL=no
IPV6INIT=no
PEERDNS=yes
[root@one-node2 ~]# more /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=one-node2
GATEWAY=192.168.25.254
```

Y lo dejamos así:

```
[root@one-node2 network-scripts]# more ifcfg-em1
DEVICE="em1"
HWADDR="D4:AE:52:7B:AF:9B"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=192.101.5.113
NETMASK=255.255.255.252
NETWORK=192.101.5.112
BROADCAST=192.101.5.115
USERCTL=no
IPV6INIT=no
PEERDNS=yes
```

Probamos a tirar y levantar el interfaz antes de hacer un reload de la configuración de networking. No queremos quedarnos sin acceso al servidor:

```
[root@one-node2 network-scripts]# ifdown em1
[root@one-node2 network-scripts]# ifup em1
[root@one-node2 network-scripts]# ifconfig em1
em1       Link encap:Ethernet  HWaddr D4:AE:52:7B:AF:9B
          inet addr:192.101.5.113  Bcast:192.101.5.115  Mask:255.255.255.252
          inet6 addr: fe80::d6ae:52ff:fe7b:af9b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1848 (1.8 KiB)  TX bytes:748 (748.0 b)
```

Bien, ahora probamos a hacer un ping a la IP del default gateway, antes de cambiarlo en el fichero network:

```
[root@one-node2 network-scripts]# ping 192.101.5.114
PING 192.101.5.114 (192.101.5.114) 56(84) bytes of data.
64 bytes from 192.101.5.114: icmp_seq=1 ttl=255 time=4.28 ms
64 bytes from 192.101.5.114: icmp_seq=2 ttl=255 time=0.719 ms
64 bytes from 192.101.5.114: icmp_seq=3 ttl=255 time=0.713 ms
64 bytes from 192.101.5.114: icmp_seq=4 ttl=255 time=0.965 ms
64 bytes from 192.101.5.114: icmp_seq=5 ttl=255 time=0.710 ms
64 bytes from 192.101.5.114: icmp_seq=6 ttl=255 time=0.698 ms
64 bytes from 192.101.5.114: icmp_seq=7 ttl=255 time=0.790 ms
64 bytes from 192.101.5.114: icmp_seq=8 ttl=255 time=0.712 ms
^C
--- 192.101.5.114 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7462ms
rtt min/avg/max/mdev = 0.698/1.198/4.280/1.168 ms
```

Ahora cambiamos el fichero network:

```
[root@one-node2 sysconfig]# more network
NETWORKING=yes
HOSTNAME=one-node2
GATEWAY=192.101.5.114
[root@one-node2 sysconfig]# service network reload
Shutting down interface em1:                               [  OK  ]
Shutting down interface em2:                               [  OK  ]
Shutting down interface em3:                               [  OK  ]
Shutting down interface em4:                               [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface em1:                                 [  OK  ]
Bringing up interface em2:                                 [  OK  ]
Bringing up interface em3:                                 [  OK  ]
Bringing up interface em4:                                 [  OK  ]
[root@one-node2 sysconfig]# ping www.cisco.com
PING e144.dscb.akamaiedge.net (23.45.192.170) 56(84) bytes of data.
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=1 ttl=55 time=50.2 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=2 ttl=55 time=50.6 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=3 ttl=55 time=50.4 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=4 ttl=55 time=50.4 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=5 ttl=55 time=50.4 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=6 ttl=55 time=50.4 ms
64 bytes from a23-45-192-170.deploy.static.akamaitechnologies.com (23.45.192.170): icmp_seq=7 ttl=55 time=50.3 ms
^C
--- e144.dscb.akamaiedge.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6637ms
rtt min/avg/max/mdev = 50.283/50.439/50.608/0.152 ms
```

Bien, vemos que tiene salida al exterior, y que resuelve correctamente por DNS.

```
[root@one-node2 sysconfig]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.101.5.112   0.0.0.0         255.255.255.252 U         0 0          0 em1
10.60.1.0       0.0.0.0         255.255.255.0   U         0 0          0 em3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em2
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em4
0.0.0.0         192.101.5.114   0.0.0.0         UG        0 0          0 em1
```

Ahora solo nos queda eliminar la línea de las iptables que permite el acceso por ssh al puerto 22 desde fuera.

```
[root@one-node2 sysconfig]# more iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2222 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
[root@one-node2 sysconfig]# service iptables restart
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
[root@one-node2 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:04:17 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6:728]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2222 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
# Completed on Tue Dec 17 18:04:17 2013
```

Nos queda hacer la prueba de acceso desde la IP pública:

```
RSA host key for IP address '192.101.5.113' not in list of known hosts.
Last login: Tue Dec 17 17:24:21 2013 from 10.134.16.105
[root@one-node2 ~]# who
root     pts/0        2013-12-17 17:22 (10.60.1.1)
root     pts/1        2013-12-17 18:07 (192.100.253.254)
[root@one-node2 ~]#
```

Con esto hemos terminado con el host2. Ahora que tenemos el host2 completamente migrado al direccionamiento público, tenemos que hacer lo mismo con el host1. Salimos de todas las sesiones abiertas en el host1 y entramos desde el host2 a través de la red privada:

```
[root@one-node2 ~]# ssh one-node1
The authenticity of host 'one-node1 (10.60.1.1)' can't be established.
RSA key fingerprint is 51:9d:de:32:55:55:eb:e7:01:19:76:5c:a2:7a:c0:a4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'one-node1,10.60.1.1' (RSA) to the list of known hosts.
root@one-node1's password:
Last login: Tue Dec 17 17:22:19 2013 from 10.134.16.105
[root@one-node1 ~]#
```

Modificamos las iptables, los puertos del servicio ssh, etc...

```
[root@one-node1 ssh]# grep ^Port sshd_config
Port 22
Port 2222
[root@one-node1 ssh]# service sshd reload
Reloading sshd:                                             [  OK  ]
[root@one-node1 ssh]# nmap -sT localhost

Starting Nmap 5.51 ( http://nmap.org ) at 2013-12-17 18:22 CET
Failed to find device em4 which was referenced in /proc/net/route
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00040s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 994 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
111/tcp  open  rpcbind
631/tcp  open  ipp
2049/tcp open  nfs
2222/tcp open  EtherNet/IP-1

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
[root@one-node1 ssh]# cd /etc/sysconfig/
[root@one-node1 sysconfig]# more iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2222 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9869 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 29876 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
[root@one-node1 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:23:45 2013
*mangle
:PREROUTING ACCEPT [44450:8745899]
:INPUT ACCEPT [39473:6897080]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [27368:4092176]
:POSTROUTING ACCEPT [27368:4092176]
COMMIT
# Completed on Tue Dec 17 18:23:45 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:23:45 2013
*nat
:PREROUTING ACCEPT [16251:4604239]
:POSTROUTING ACCEPT [1489:89756]
:OUTPUT ACCEPT [1489:89756]
COMMIT
# Completed on Tue Dec 17 18:23:45 2013
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:23:45 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [27368:4092176]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9869 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 29876 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Dec 17 18:23:45 2013
[root@one-node1 sysconfig]# service iptables restart
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: mangle nat filte[  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
```

```
[root@one-node1 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:25:37 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [30:3962]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2222 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9869 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 29876 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
# Completed on Tue Dec 17 18:25:37 2013
```

Ahora cambiamos el interfaz de salida:

```
[root@one-node1 network-scripts]# more ifcfg-em1
DEVICE="em1"
HWADDR="D4:AE:52:7B:AF:34"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=192.168.25.203
NETMASK=255.255.255.0
NETWORK=192.168.25.0
BROADCAST=192.168.25.255
USERCTL=no
IPV6INIT=no
PEERDNS=yes
```

Lo dejamos así:

```
[root@one-node1 network-scripts]# more ifcfg-em1
DEVICE="em1"
HWADDR="D4:AE:52:7B:AF:34"
NM_CONTROLLED="no"
ONBOOT="yes"
IPADDR=192.101.5.109
NETMASK=255.255.255.252
NETWORK=192.101.5.108
BROADCAST=192.101.5.111
USERCTL=no
IPV6INIT=no
PEERDNS=yes
```

```
[root@one-node1 sysconfig]# more network
NETWORKING=yes
HOSTNAME=one-node1
GATEWAY=192.101.5.110
```

Probamos a tirar y levantar el interface:

```
[root@one-node1 network-scripts]# ifdown em1
[root@one-node1 network-scripts]# ifup em1
[root@one-node1 network-scripts]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.101.5.108   0.0.0.0         255.255.255.252 U         0 0          0 em1
10.60.1.0       0.0.0.0         255.255.255.0   U         0 0          0 em3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em2
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 em4
0.0.0.0         192.101.5.110   0.0.0.0         UG        0 0          0 em1
```

Perfecto, ya lo tenemos. Vamos a probarlo hacia afuera:

```
[root@one-node1 network-scripts]# ping www.cisco.com
PING e144.dscb.akamaiedge.net (23.77.112.170) 56(84) bytes of data.
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=1 ttl=54 time=35.2 ms
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=2 ttl=54 time=35.2 ms
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=3 ttl=54 time=35.1 ms
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=4 ttl=54 time=35.2 ms
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=5 ttl=54 time=35.2 ms
64 bytes from a23-77-112-170.deploy.static.akamaitechnologies.com (23.77.112.170): icmp_seq=6 ttl=54 time=35.2 ms
^C
--- e144.dscb.akamaiedge.net ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5360ms
rtt min/avg/max/mdev = 35.180/35.233/35.278/0.219 ms
```

Ahora nos queda eliminar la línea de acceso al puerto 22 por ssh desde afuera en las iptables:

```
[root@one-node1 sysconfig]# more iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2222 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9869 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 29876 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
[root@one-node1 sysconfig]# service iptables restart
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
[root@one-node1 sysconfig]# iptables-save
# Generated by iptables-save v1.4.7 on Tue Dec 17 18:40:33 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [84:14673]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 10.60.1.0/24 -i em3 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2222 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9869 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 29876 -j ACCEPT
-A INPUT -j DROP
-A FORWARD -m physdev --physdev-is-bridged -j ACCEPT
-A FORWARD -j DROP
COMMIT
# Completed on Tue Dec 17 18:40:33 2013
```

Y entramos a la máquina desde fuera:

```
[jamontes@Tesla ~]$  ssh root@one-node1 -p 2222
RSA host key for IP address '192.101.5.109' not in list of known hosts.
Last login: Tue Dec 17 18:13:31 2013 from 10.60.1.2
[root@one-node1 ~]#
```

Con esto damos la migración de los dos servidores con direccionamiento 100% público por concluida.

4. Preparación de la plataforma para pruebas y producción

Ahora vamos a trabajar sobre el entorno de producción, para preparar el datastore, las imágenes, las redes, las cuentas de usuario, etc...

Se trata de ponerlo 100% en producción, tras realizar las pruebas de rendimiento y completar la configuración.

4.1. Trabajando con el datastore qcow y preparando las imágenes

Vamos a empezar por el datastore. Vamos a crear un datastore de tipo qcow2, para poder sacarle partido a las instancias contextualizadas.

Creamos la plantilla para generar el datastore:

```
[oneadmin@one-node1 mytemplates]$ more datastore_qcow.tmpl
NAME = qcow2_images
DS_MAD = fs
TM_MAD = qcow2
```

Creamos el datastore:

```
[oneadmin@one-node1 mytemplates]$ onedatastore list
  ID NAME                SIZE AVAIL CLUSTER      IMAGES TYPE DS       TM
   0 system                - -     -                 0 sys  -        shared
   1 default              2T 94%   -                 4 img  fs       shared
   2 files                2T 94%   -                 0 fil  fs       ssh
[oneadmin@one-node1 mytemplates]$ onedatastore create datastore_qcow.tmpl
ID: 100
[oneadmin@one-node1 mytemplates]$ onedatastore list
  ID NAME                SIZE AVAIL CLUSTER      IMAGES TYPE DS       TM
   0 system                - -     -                 0 sys  -        shared
   1 default              2T 94%   -                 4 img  fs       shared
   2 files                2T 94%   -                 0 fil  fs       ssh
 100 qcow2_images         2T 94%   -                 0 img  fs       qcow2
[oneadmin@one-node1 mytemplates]$ onedatastore show 100
DATASTORE 100 INFORMATION
ID             : 100
NAME           : qcow2_images
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : IMAGE
DS_MAD         : fs
TM_MAD         : qcow2
BASE PATH      : /var/lib/one/datastores/100
DISK_TYPE      : FILE

DATASTORE CAPACITY
TOTAL:         : 2T
USED:          : 1M
FREE:          : 1.8T

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---
```

```
DATASTORE TEMPLATE
DS_MAD="fs"
TM_MAD="qcow2"

IMAGES
```

Bien, ahora tenemos que importar las imágenes que hemos contextualizado en el nuevo datastore.

```
[oneadmin@one-node1 mytemplates]$ more centos65_x86_64_qcow2.tmpl
NAME          = "CentOS 6.5 non persistent"
PATH          = /home/datastores/centos64_x86_64.qcow2
TYPE          = OS
DRIVER        = qcow2
DESCRIPTION   = "CentOS 6.5 64 bits with qcow2 driver non persistent"
[oneadmin@one-node1 mytemplates]$ oneimage create -d qcow2_images centos65_x86_64_qcow2.tmpl
ID: 5
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default        11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default        10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
[oneadmin@one-node1 mytemplates]$ oneimage show 5
IMAGE 5 INFORMATION
ID             : 5
NAME           : CentOS 6.5 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 11:44:15
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/d660bcd3f0d64b37634e000e40d25238
PATH           : /home/datastores/centos64_x86_64.qcow2
SIZE           : 3.2G
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
```

```
DESCRIPTION="CentOS 6.5 64 bits with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES

```

Habilitamos los permisos para poder utilizar la imagen:

```
[oneadmin@one-node1 mytemplates]$ oneimage chmod 5 644
[oneadmin@one-node1 mytemplates]$ oneimage show 5
IMAGE 5 INFORMATION
ID             : 5
NAME           : CentOS 6.5 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 11:44:15
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/d660bcd3f0d64b37634e000e40d25238
PATH           : /home/datastores/centos64_x86_64.qcow2
SIZE           : 3.2G
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="CentOS 6.5 64 bits with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES

```

Ahora importamos la imagen tty_linuxV2 en los dos datastores: como imagen raw no persistente, y como imagen qcow2 no persistente. Creamos la imagen qcow2:

```
[root@one-node1 ~]# cd /home/datastores/
[root@one-node1 datastores]# ll
total 4165192
-rw-r--r-- 1 root     root      796524544 Sep 25 15:41 c6-x86_64-20130910-1.qcow2
-rw-r--r-- 1 root     root     3426680832 Dec  5 13:42 centos64_x86_64.qcow2
-rw-r--r-- 1 oneadmin oneadmin   41943040 Dec 16 18:04 tty_linuxV2.img
```

```
[root@one-node1 datastores]# qemu-img info tty_linuxV2.img
image: tty_linuxV2.img
file format: raw
virtual size: 40M (41943040 bytes)
disk size: 40M
[root@one-node1 datastores]# qemu-img convert -O qcow2 ./tty_linuxV2.img ./tty_linuxV2.qcow2
[root@one-node1 datastores]# ll
total 4175760
-rw-r--r-- 1 root     root      796524544 Sep 25 15:41 c6-x86_64-20130910-1.qcow2
-rw-r--r-- 1 root     root     3426680832 Dec  5 13:42 centos64_x86_64.qcow2
-rw-r--r-- 1 oneadmin oneadmin   41943040 Dec 16 18:04 tty_linuxV2.img
-rw-r--r-- 1 root     root       10944512 Dec 19 11:53 tty_linuxV2.qcow2
[root@one-node1 datastores]# qemu-img info tty_linuxV2.qcow2
image: tty_linuxV2.qcow2
file format: qcow2
virtual size: 40M (41943040 bytes)
disk size: 10M
cluster_size: 65536
```

Ahora creamos las dos plantillas para los dos datastores:

```
[oneadmin@one-node1 mytemplates]$ more tty_linuxV2_qcow2.tmpl tty_linuxV2_raw.tmpl
::::::::::::::
tty_linuxV2_qcow2.tmpl
::::::::::::::
NAME          = "tty_linuxV2 qcow2 non persistent"
PATH          = /home/datastores/tty_linuxV2.qcow2
TYPE          = OS
DRIVER        = qcow2
DESCRIPTION   = "tty_linuxV2 contextualized with qcow2 driver non persistent"
::::::::::::::
tty_linuxV2_raw.tmpl
::::::::::::::
NAME          = "tty_linuxV2 raw non persistent"
PATH          = /home/datastores/tty_linuxV2.img
TYPE          = OS
DRIVER        = raw
DESCRIPTION   = "tty_linuxV2 contextualized with raw driver non persistent"
```

E importamos las imágenes en los datastores correspondientes:

```
[oneadmin@one-node1 mytemplates]$ oneimage create -d qcow2_images tty_linuxV2_qcow2.tmpl
ID: 6
[oneadmin@one-node1 mytemplates]$ oneimage create -d default tty_linuxV2_raw.tmpl
ID: 7
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default        11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default        10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
   6 oneadmin   oneadmin   tty_linuxV2 qco qcow2_imag     11M OS    No rdy     0
   7 oneadmin   oneadmin   tty_linuxV2 raw default        40M OS    No rdy     0
[oneadmin@one-node1 mytemplates]$ oneimage show 6
IMAGE 6 INFORMATION
ID             : 6
NAME           : tty_linuxV2 qcow2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 12:19:22
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/468370a07592933b3ab97b2e9958b135
PATH           : /home/datastores/tty_linuxV2.qcow2
SIZE           : 11M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV2 contextualized with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ oneimage show 7
IMAGE 7 INFORMATION
ID             : 7
NAME           : tty_linuxV2 raw non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 12/19 12:19:47
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/ca5a240f355ce8b4fed33f77946f47dc
PATH           : /home/datastores/tty_linuxV2.img
```

```
SIZE           : 40M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV2 contextualized with raw driver non persistent"
DEV_PREFIX="hd"
DRIVER="raw"

VIRTUAL MACHINES
```

Ahora habilitamos los permisos de uso de las imágenes:

```
[oneadmin@one-node1 mytemplates]$ oneimage chmod 6 644
[oneadmin@one-node1 mytemplates]$ oneimage chmod 7 644
[oneadmin@one-node1 mytemplates]$ oneimage show 6
IMAGE 6 INFORMATION
ID             : 6
NAME           : tty_linuxV2 qcow2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 12:19:22
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/468370a07592933b3ab97b2e9958b135
PATH           : /home/datastores/tty_linuxV2.qcow2
SIZE           : 11M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV2 contextualized with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"
```

```
VIRTUAL MACHINES


[oneadmin@one-node1 mytemplates]$ oneimage show 7
IMAGE 7 INFORMATION
ID             : 7
NAME           : tty_linuxV2 raw non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 12/19 12:19:47
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/ca5a240f355ce8b4fed33f77946f47dc
PATH           : /home/datastores/tty_linuxV2.img
SIZE           : 40M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV2 contextualized with raw driver non persistent"
DEV_PREFIX="hd"
DRIVER="raw"

VIRTUAL MACHINES
```

Vamos a dejar disponibles algunas imágenes más, que serán necesarias para los usuarios:

```
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default        11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default        10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
   6 oneadmin   oneadmin   tty_linuxV2 qco qcow2_imag     11M OS    No rdy     0
   7 oneadmin   oneadmin   tty_linuxV2 raw default        40M OS    No rdy     0
```

Vamos a migrar el router (id 1), la versión de CentOS 6.2 (id 4), y la de Ubuntu server (id 3) al datastore qcow, para que las imágenes consuman menos y sea más rápido lanzarlas. Tenemos que migrar las imágenes a formato qcow2. Vamos una a una:

```
[oneadmin@one-node1 mytemplates]$ oneimage show  1
IMAGE 1 INFORMATION
ID              : 1
NAME            : OpenNebula 4.2 Virtual Router
USER            : oneadmin
GROUP           : oneadmin
DATASTORE       : default
TYPE            : OS
REGISTER TIME   : 11/15 01:44:16
PERSISTENT      : No
SOURCE          : /var/lib/one/datastores/1/e83b66bcb88a4f7be1cd41c9d16bb314
PATH            : http://marketplace.c12g.com/appliance/51f2a09f8fb81d4d19000004/download
SIZE            : 83M
STATE           : rdy
RUNNING_VMS     : 0

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

IMAGE TEMPLATE
DESCRIPTION="Virtual Router Appliance"
DEV_PREFIX="hd"
DRIVER="raw"
MD5="78d46f5516c08e0d96a8dc92aa26c838"
SHA1="a2a538027d5f9f9fcbbad6c8adad3f67d2de5242"

VIRTUAL MACHINES
[root@one-node1 datastores]# qemu-img convert -O qcow2
/var/lib/one/datastores/1/e83b66bcb88a4f7be1cd41c9d16bb314 virtual_router.qcow2
[root@one-node1 datastores]# qemu-img info virtual_router.qcow2
image: virtual_router.qcow2
file format: qcow2
virtual size: 200M (209715200 bytes)
disk size: 190M
cluster_size: 65536
```

Ahora creamos la plantilla para importar la imagen:

```
[oneadmin@one-node1 mytemplates]$ more virtual_router_qcow2.tmpl
NAME           = "virtual_router qcow2 non persistent"
PATH           = /home/datastores/virtual_router.qcow2
TYPE           = OS
DRIVER         = qcow2
DESCRIPTION    = "virtual_router with qcow2 driver non persistent"
```

```
[oneadmin@one-node1 mytemplates]$ oneimage create -d qcow2_images virtual_router_qcow2.tmpl
ID: 8
[oneadmin@one-node1 mytemplates]$ oneimage chmod 8 644
[oneadmin@one-node1 mytemplates]$ oneimage show 8
IMAGE 8 INFORMATION
ID             : 8
NAME           : virtual_router qcow2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 13:09:39
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/cdea302029b725e74ed2e69166858d11
PATH           : /home/datastores/virtual_router.qcow2
SIZE           : 191M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="virtual_router with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ oneimage show 3
IMAGE 3 INFORMATION
ID             : 3
NAME           : Ubuntu Server 12.04 (Precise Pangolin) - kvm
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 11/16 00:29:42
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/644079a4862479b8e0c26279acd1f058
PATH           : http://marketplace.c12g.com/appliance/4fc76a938fb81d3517000001/download
SIZE           : 11G
STATE          : rdy
```

```
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DESCRIPTION="Ubuntu Server 12.04 LTS"
DEV_PREFIX="hd"
DRIVER="raw"
MD5="80d4bb65e41335df051a9d5d43c3e90d"

VIRTUAL MACHINES

[root@one-node1 datastores]# qemu-img convert -O qcow2
/var/lib/one/datastores/1/644079a4862479b8e0c26279acd1f058 ubuntu_server_1204.qcow2
[root@one-node1 datastores]# qemu-img info ubuntu_server_1204.qcow2
image: ubuntu_server_1204.qcow2
file format: qcow2
virtual size: 11G (11811160064 bytes)
disk size: 589M
cluster_size: 65536
[oneadmin@one-node1 mytemplates]$ more ubuntu_server_qcow2.tmpl
NAME          = "Ubuntu Server 12.04 LTS qcow2 non persistent"
PATH          = /home/datastores/ubuntu_server_1204.qcow2
TYPE          = OS
DRIVER        = qcow2
DESCRIPTION   = "Ubuntu Server 12.04 LTS contextualized with qcow2 driver non persistent"
[oneadmin@one-node1 mytemplates]$ oneimage chmod 9 644
[oneadmin@one-node1 mytemplates]$ oneimage show 9
IMAGE 9 INFORMATION
ID             : 9
NAME           : Ubuntu Server 12.04 LTS qcow2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 13:18:54
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/2da986e322b259ae9df15328d1c0deaa
PATH           : /home/datastores/ubuntu_server_1204.qcow2
SIZE           : 589M
STATE          : rdy
RUNNING_VMS    : 0
```

```
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="Ubuntu Server 12.04 LTS contextualized with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ oneimage show 4
IMAGE 4 INFORMATION
ID             : 4
NAME           : CentOS Server 6.2
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 11/16 00:45:25
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/9a11234cab1c05898e22b048b157cd83
PATH           : http://marketplace.c12g.com/appliance/4fc76a938fb81d3517000002/download
SIZE           : 10G
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DESCRIPTION="CentOS Server 6.2"
DEV_PREFIX="hd"
DRIVER="raw"
MD5="94176fbba17f5efcabb0b05fc55b1a16"

VIRTUAL MACHINES



[root@one-node1 datastores]# qemu-img convert -O qcow2
/var/lib/one/datastores/1/9a11234cab1c05898e22b048b157cd83 centos62_x86_64.qcow2
```

```
[root@one-node1 datastores]# qemu-img info centos62_x86_64.qcow2
image: centos62_x86_64.qcow2
file format: qcow2
virtual size: 10G (10737418240 bytes)
disk size: 1.7G
cluster_size: 65536
[oneadmin@one-node1 mytemplates]$ more centos62_x86_64_qcow2.tmpl
NAME          = "CentOS 6.2 non persistent"
PATH          = /home/datastores/centos62_x86_64.qcow2
TYPE          = OS
DRIVER        = qcow2
DESCRIPTION   = "CentOS 6.2 64 bits with qcow2 driver non persistent"
[oneadmin@one-node1 mytemplates]$ oneimage create -d qcow2_images centos62_x86_64_qcow2.tmpl
ID: 10
[oneadmin@one-node1 mytemplates]$ oneimage chmod 10 644
[oneadmin@one-node1 mytemplates]$ oneimage show 10
IMAGE 10 INFORMATION
ID             : 10
NAME           : CentOS 6.2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/19 13:25:17
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/5b11573d6e415ae73ca9f7c732c8d9d5
PATH           : /home/datastores/centos62_x86_64.qcow2
SIZE           : 1.8G
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="CentOS 6.2 64 bits with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES
```

4.2. Preparando las redes de uso para los usuarios

Bien, ahora tenemos que crear las plantillas de las redes que podrán utilizar los usuarios. Después de la migración a IPs públicas, tenemos que dar de alta la plantilla de la red pública directa.

```
[oneadmin@one-node1 mytemplates]$ more public_lan.net
NAME = "Internet"
TYPE = "RANGED"
PHYDEV = "em2"
BRIDGE = "brp0"
VLAN = NO

NETWORK_ADDRESS = "192.101.5.32/27"
GATEWAY = "192.101.5.62"
DNS = "8.8.8.8 8.8.4.4"
IP_START = "192.101.5.33"
IP_END = "192.101.5.61"
[oneadmin@one-node1 mytemplates]$ onevnet create public_lan.net
ID: 2
[oneadmin@one-node1 mytemplates]$ onevnet chmod 2 644
[oneadmin@one-node1 mytemplates]$ onevnet show 2
VIRTUAL NETWORK 2 INFORMATION
ID             : 2
NAME           : Internet
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brp0
VLAN           : No
PHYSICAL DEVICE: em2
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.101.5.62"
NETWORK_ADDRESS="192.101.5.32/27"
NETWORK_MASK="255.255.255.224"

RANGE
IP_START       : 192.101.5.33
```

```
IP_END         : 192.101.5.61

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   0 oneadmin     oneadmin     Internet NAT LA -               R virbr0        0
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          0
```

La red 0 ya no tiene sentido usarla, porque la hemos desactivado para evitar pasar tráfico a través de los interfaces de gestión de los hosts. Pero para borrarla tenemos que eliminar también la plantilla que utiliza esa red.

```
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER            GROUP           NAME                                    REGTIME
   0 oneadmin        oneadmin        tty test 2 NICs 1            11/16 01:29:12
[oneadmin@one-node1 mytemplates]$ onetemplate show 0
TEMPLATE 0 INFORMATION
ID             : 0
NAME           : tty test 2 NICs 1
USER           : oneadmin
GROUP          : oneadmin
REGISTER TIME  : 11/16 01:29:12

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CPU="0.1"
DISK=[
  IMAGE_ID="0" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="Internet NAT LAN" ]
NIC=[
  NETWORK="VLAN2" ]
```

Así que empezamos borrando la plantilla de la máquina virtual, y a continuación la de la red con NAT. La idea es que cualquier salida a internet se realice a través de un router virtual que gestione una VLAN privada, y consuma una única IP pública por VLAN.

```
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER             GROUP           NAME                                REGTIME
   0 oneadmin         oneadmin        tty test 2 NICs 1            11/16 01:29:12
[oneadmin@one-node1 mytemplates]$ onetemplate delete 0
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER             GROUP           NAME                                REGTIME

[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   0 oneadmin     oneadmin     Internet NAT LA -               R virbr0        0
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          0
[oneadmin@one-node1 mytemplates]$ onevnet delete 0
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          0
```

4.3. Trabajando con cuentas de usuario

Vamos a crear una cuenta de usuario, que llamaremos "customer1" con password "1solomio"

```
[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME             GROUP      AUTH            VMS            MEMORY         CPU
   0 oneadmin         oneadmin   core              -                 -           -
   1 serveradmin      oneadmin   server_c          -                 -           -
[oneadmin@one-node1 mytemplates]$ oneuser create customer1 1solomio
ID: 2
[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME             GROUP      AUTH            VMS            MEMORY         CPU
   0 oneadmin         oneadmin   core              -                 -           -
   1 serveradmin      oneadmin   server_c          -                 -           -
   2 customer1        users      core              -                 -           -
[oneadmin@one-node1 mytemplates]$ oneuser show 2
USER 2 INFORMATION
ID              : 2
NAME            : customer1
GROUP           : users
PASSWORD        : 9e8ce8b24a55028eb4cad409375f107a4700a781
AUTH_DRIVER     : core
ENABLED         : Yes

USER TEMPLATE
TOKEN_PASSWORD="67e9aedafe4fc2280283d96364bbc0561a034490"

RESOURCE USAGE & QUOTAS
```

Ahora vamos a crearle unas cuantas plantillas para que pueda hacer uso de las máquinas y los sistemas. Esta es la plantilla que usaremos en las máquinas virtuales:

```
[oneadmin@one-node1 mytemplates]$ more priv_vlan3.net
NAME = "VLAN3"
TYPE = "RANGED"
PHYDEV = "em2"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brhm3"

NETWORK_ADDRESS = "192.168.120.0/21"
GATEWAY = "192.168.120.1"
DNS = "8.8.8.8 8.8.4.4"
IP_START = "192.168.120.1"
IP_END = "192.168.127.254"
```

Cambiamos al propietario de la red:

```
[oneadmin@one-node1 mytemplates]$ onevnet chown 3 2 1
[oneadmin@one-node1 mytemplates]$ onevnet show 3
VIRTUAL NETWORK 3 INFORMATION
ID             : 3
NAME           : VLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brhm3
VLAN           : Yes
PHYSICAL DEVICE: em2
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254
```

```
VIRTUAL MACHINES
```

4.3.1. Trabajando con el router virtual para el usuario customer1

Ahora creamos la plantilla de red para el router que gestionará la VLAN3:

```
[oneadmin@one-node1 mytemplates]$ more router_vlan3.net
NAME = "RTVLAN3"
TYPE = "RANGED"
PHYDEV = "em2"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brhm3"

NETWORK_ADDRESS = "192.168.120.0/21"
[oneadmin@one-node1 mytemplates]$ onevnet create router_vlan3.net
ID: 4
[oneadmin@one-node1 mytemplates]$ onevnet chown 4 2 1
[oneadmin@one-node1 mytemplates]$ onevnet show 4
VIRTUAL NETWORK 4 INFORMATION
ID              : 4
NAME            : RTVLAN3
USER            : customer1
GROUP           : users
CLUSTER         : -
TYPE            : RANGED
BRIDGE          : brhm3
VLAN            : Yes
PHYSICAL DEVICE: em2
VLAN ID         : 3
USED LEASES     : 0

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START        : 192.168.120.1
IP_END          : 192.168.127.254

VIRTUAL MACHINES
```

Ahora creamos la plantilla para la instancia router de la VLAN3. Esta plantilla tendrá permisos totales por parte del usuario para modificarla según sus necesidades:

```
[oneadmin@one-node1 mytemplates]$ more router_vlan3.tmpl
NAME="RouterVLAN3"
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
FEATURES=[
  ACPI="yes" ]

NIC = [ NETWORK="Internet" ]
NIC = [
  NETWORK="RTVLAN3",
  IP="192.168.120.1" ]

CONTEXT=[
  TARGET         = "hdb",
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  PRIVNET        = "$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET         = "$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  TEMPLATE       = "$TEMPLATE",
  DHCP           = "NO",
  RADVD          = "NO",
  FORWARDING     = "2222:192.168.120.2:22" ]
[oneadmin@one-node1 mytemplates]$ onetemplate create router_vlan3.tmpl
ID: 1
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER            GROUP           NAME                                REGTIME
   1 oneadmin        oneadmin        RouterVLAN3                  12/19 16:21:49
[oneadmin@one-node1 mytemplates]$ onetemplate chown 1 2 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 1
TEMPLATE 1 INFORMATION
ID             : 1
```

```
NAME            : RouterVLAN3
USER            : customer1
GROUP           : users
REGISTER TIME   : 12/19 16:21:49

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  DHCP="NO",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",
  PRIVNET="$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET="$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",
  TEMPLATE="$TEMPLATE" ]
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="Internet" ]
NIC=[
  IP="192.168.120.1",
  NETWORK="RTVLAN3" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Ya lo tenemos preparado. Vamos a probar a instanciarlo:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 1 --name "RouterVLAN7" --user customer1
--password 1solomio
VM ID: 212
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   212 customer users    RouterVLAN7     fail    0      0K               0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm show 212
VIRTUAL MACHINE 212 INFORMATION
ID                  : 212
NAME                : RouterVLAN7
USER                : customer1
GROUP               : users
STATE               : FAILED
LCM_STATE           : LCM_INIT
RESCHED             : No
START TIME          : 12/19 16:47:46
END TIME            : 12/19 16:48:03
DEPLOY ID           : -

VIRTUAL MACHINE MONITORING
USED MEMORY         : 0K
NET_RX              : 0K
USED CPU            : 0
NET_TX              : 0K

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    virtual_router qcow2 non persistent file   NO       -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet               no brp0         192.101.5.33    02:00:d4:e7:05:21
                                           fe80::400:d4ff:fee7:521
  1 RTVLAN3               yes brhm3        192.168.120.1   02:00:c0:a8:78:01
                                           fe80::400:c0ff:fea8:7801

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START        TIME      PROLOG
  0 one-node2       none             erro  12/19 16:48:02   0d 00h00m   0h00m00s
```

```
USER TEMPLATE
ERROR="Thu Dec 19 16:48:03 2013 : Error deploying virtual machine: Could not create domain from
/var/lib/one//datastores/0/212/deployment.0"

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DHCP="NO",
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.101.5.62",
  ETH0_IP="192.101.5.33",
  ETH0_MASK="255.255.255.224",
  ETH0_NETWORK="192.101.5.32/27",
  ETH1_IP="192.168.120.1",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="192.168.120.0/21",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",

PRIVNET="PFZORVQ+PElEPjQ8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkFN
RT51c2VyczwvR05BTUU+PE5BTUU+UlRWTEFOMzwvTkFNRT48UEVSTUlTU0lPTlM+PE9XTkVSX1U+MTwvT1dORVJfVT48T1dORVJfT
T4xPC9PV05FUl9NPjxPV05FUl9BPjA8L09XTkVSX0E+PEdST1VQX1U+MDwvR1JPVVBfVT48R1JPVVBfTT4wPC9HUk9VUF9NPjxHUk
9VUF9BPjA8L0dST1VQX0E+PE9USEVSX1U+MDwvT1RIRVJfVT48T1RIRVJfTT4wPC9PVEhFUl9NPjxPVEhFUl9BPjA8L09USEVSX0E
+PC9QRVJNSVNTSU9OUz48Q0xVU1RFUl9JRD4tMTwvQ0xVU1RFUl9JRD48Q0xVU1RFUj48L0NMVVNURVI+PFRZUEU+MDwvVFlQRT48
QlJJREdFPmJyaG0zPC9CUklER0U+PFZMQU4+MTwvVkxBTj48UEhZREVWPmVtMjwvUEhZREVWPjxWTEFOX0lEPjM8L1ZMQU5fSUQ+P
EdMT0JBTF9QUkVGSVgvPjxTSVRFX1BSRUZJWC8+PFJBTkdFPjxJUF9TVEFSVD4xOTIuMTY4LjEyMC4xPC9JUF9TVEFSVD48SVBfRU
5EPjE5Mi4xNjguMTI3LjI1NDwvSVBfRU5EPjwvUkFOR0U+PFRPVEFMX0xFQVNFUz4xPC9UT1RBTF9MRUFTRVM+PFRFTVBMQVRFPjx
ORVRXT1JLX0FERFJFU1M+PCFbQ0RBVEFbMTkyLjE2OC4xMjAuMC8yMV1dPjwvTkVUV09SS19BRERSRVNTPjxORVRXT1JLX01BU0s+
PCFbQ0RBVEFbMjU1LjI1NS4yNDguMF1dPjwvTkVUV09SS19NQVNLPjwvVEVNUExBVEU+PExFQVNFUz48TEVBU0U+PE1BQz4wMjowM
DpjMDphODo3ODowMTwvTUFDPjxJUD4xOTIuMTY4LjEyMC4xPC9JUD48SVA2X0xJTks+ZmU4MDo6NDAwOmMwZmY6ZmVhODo3ODAxPC
9JUDZfTElOSz48VVNFRD4xPC9VU0VEPjxWSUQ+MjEyPC9WSUQ+PC9MRUFTRT48L0xFQVNFUz48L1ZORVQ+",

PUBNET="PFZORVQ+PElEPjI8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkFNR
T51c2VyczwvR05BTUU+PE5BTUU+SW50ZXJuZXQ8L05BTUU+PFBFUk1JU1NJT05TPjxPV05FUl9VPjE8L09XTkVSX1U+PE9XTkVSX0
0+MTwvT1dORVJfTT48T1dORVJfQT4wPC9PV05FUl9BPjxHUk9VUF9VPjE8L0dST1VQX1U+PEdST1VQX00+MTwvR1JPVVBfTT48R1J
PVVBfQT4wPC9HUk9VUF9BPjxPVEhFUl9VPjE8L09USEVSX1U+PE9USEVSX00+MTwvT1RIRVJfTT48T1RIRVJfQT4wPC9PVEhFUl9B
PjwvUEVSTUlTU0lPTlM+PENMVVNURVJfSUQ+LTE8L0NMVVNURVJfSUQ+PENMVVNURVI+PC9DTFVTVEVSPjxUWVBFPjA8L1RZUEU+P
EJSSURHRT5icnAwPC9CUklER0U+PFZMQU4+MDwvVkxBTj48UEhZREVWPmVtMjwvUEhZREVWPjxWTEFOX0lELz48R0xPQkFMX1BSRU
ZJWC8+PFNJVEVfUFJFRklYLz48UkFOR0U+PElQX1NUQVJUPjIxMi4yMzEuNS4zMzwvSVBfU1RBUlQ+PElQX0VORD4yMTIuMjMxLjU
uNjE8L0lQX0VORD48L1JBTkdFPjxUT1RBTF9MRUFTRVM+MTwvVE9UQUxfTEVBU0VTPjxURU1QTEFURT48RE5TPjwhW0NEQVRBWzgu
OC44LjggOC44LjQuNF1dPjwvRE5TPjxHQVRFV0FZPjwhW0NEQVRBWzIxMi4yMzEuNS42Ml1dPjwvR0FURVdBWT48TkVUV09SS19BR
ERSRVNTPjwhW0NEQVRBWzIxMi4yMzEuNS4zMi8yN11dPjwvTkVUV09SS19BRERSRVNTPjxORVRXT1JLX01BU0s+PCFbQ0RBVEFbMj
U1LjI1NS4yNTUuMjI0XV0+PC9ORVRXT1JLX01BU0s+PC9URU1QTEFURT48TEVBU0VTPjxMRUFTRT48TUFDPjAyOjAwOmQ0OmU3OjA
1OjIxPC9NQUM+PElQPjIxMi4yMzEuNS4zMzwvSVA+PElQNl9MSU5LPmZlODA6OjQwMDpkNGZmOmZlZTc6NTIxPC9JUDZfTElOSz48
VVNFRD4xPC9VU0VEPjxWSUQ+MjEyPC9WSUQ+PC9MRUFTRT48L0xFQVNFUz48L1ZORVQ+",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",
TEMPLATE="PFZNPjxJRD4yMTI8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkF
NRT51c2VyczwvR05BTUU+PE5BTUU+Um91dGVyVkxBTjc8L05BTUU+PFBFUk1JU1NJT05TPjxPV05FUl9VPjE8L09XTkVSX1U+PE9X
TkVSX00+MTwvT1dORVJfTT48T1dORVJfQT4wPC9PV05FUl9BPjxHUk9VUF9VPjA8L0dST1VQX1U+PEdST1VQX00+MDwvR1JPVVBfT
T48R1JPVVBfQT4wPC9HUk9VUF9BPjxPVEhFUl9VPjA8L09USEVSX1U+PE9USEVSX00+MDwvT1RIRVJfTT48T1RIRVJfQT4wPC9PVE
hFUl9BPjwvUEVSTUlTU0lPTlM+PExBU1RfUE9MTD4wPC9MQVNUX1BPTEw+PFNUQVRFPjE8L1NUQVRFPjxMQ01fU1RBVEU+MDwvTEN
NX1NUQVRFPjxSRVNDSEVEPjA8L1JFU0NIRUQ+PFNUSU1FPjEzODc0NjgwNjY8L1NUSU1FPjxFVElNRT4wPC9FVElNRT48REVQTE9Z
X0lEPjwvREVQTE9ZX0lEPjxNRU1PUlk+MDwvTUVNT1JZPjxDUFU+MDwvQ1BVPjxORVRfVFg+MDwvTkVUX1RYPjxORVRfUlg+MDwvT
kVUX1JYPjxURU1QTEFURT48Q1BVPjwhW0NEQVRBWzAuMl1dPjwvQ1BVPjxESVNLPjxDTE9ORT48IVtDREFUQVtZRVNdXT48L0NMT0
5FPjxEQVRBU1RPUkU+PCFbQ0RBVEFbcWNvdzJfaW1hZ2VzXV0+PC9EQVRBU1RPUkU+PERBVEFTVE9SRV9JRD48IVtDREFUQVsxMDB
dXT48L0RBVEFTVE9SRV9JRD48REVWX1BSRUZJWD48IVtDREFUQVtoZF1dPjwvREVWX1BSRUZJWD48RElTS19JRD48IVtDREFUQVsw
XV0+PC9ESVNLX0lEPjxEUklWRVI+PCFbQ0RBVEFbcWNvdzJdXT48L0RSSVZFUj48SU1BR0U+PCFbQ0RBVEFbdmlydHVhbF9yb3V0Z
XIgcWNvdzIgbm9uIHBlcnNpc3RlbnRdXT48L0lNQUdFPjxJTUFHRV9JRD48IVtDREFUQVs4XV0+PC9JTUFHRV9JRD48UkVBRE9OTF
k+PCFbQ0RBVEFbTk9dXT48L1JFQURPTkxZPjxTQVZFPjwhW0NEQVRBW05PXV0+PC9TQVZFPjxTT1VSQ0U+PCFbQ0RBVEFbL3Zhci9
saWIvb25lL2RhdGFzdG9yZXMvMTAwL2NkZWEzMDIwMjliNzI1ZTc0ZWQyZTY5MTY2ODU4ZDExXV0+PC9TT1VSQ0U+PFRBUkdFVD48
IVtDREFUQVtoZGFdXT48L1RBUkdFVD48VE1fTUFEPjwhW0NEQVRBW3Fjb3cyXV0+PC9UTV9NQUQ+PFRZUEU+PCFbQ0RBVEFbRklMR
V1dPjwvVFlQRT48L0RJU0s+PE1FTU9SWT48IVtDREFUQVs2NF1dPjwvTUVNT1JZPjxOSUM+PEJSSURHRT48IVtDREFUQVticnAwXV
0+PC9CUklER0U+PElQPjwhW0NEQVRBWzIxMi4yMzEuNS4zM11dPjwvSVA+PElQNl9MSU5LPjwhW0NEQVRBW2ZlODA6OjQwMDpkNGZ
mOmZlZTc6NTIxXV0+PC9JUDZfTElOSz48TUFDPjwhW0NEQVRBWzAyOjAwOmQ0OmU3OjA1OjIxXV0+PC9NQUM+PE5FVFdPUks+PCFb
Q0RBVEFbSW50ZXJuZXRdXT48L05FVFdPUks+PE5FVFdPUktfSUQ+PCFbQ0RBVEFbMl1dPjwvTkVUV09SS19JRD48TklDX0lEPjwhW
0NEQVRBWzBdXT48L05JQ19JRD48UEhZREVWPjwhW0NEQVRBW2VtMl1dPjwvUEhZREVWPjxWTEFOPjwhW0NEQVRBW05PXV0+PC9WTE
FOPjwvTklDPjxOSUM+PEJSSURHRT48IVtDREFUQVticmhtM11dPjwvQlJJREdFPjxJUD48IVtDREFUQVsxOTIuMTY4LjEyMC4xXV0
+PC9JUD48SVA2X0xJTks+PCFbQ0RBVEFbZmU4MDo6NDAwOmMwZmY6ZmVhODo3ODAxXV0+PC9JUDZfTElOSz48TUFDPjwhW0NEQVRB
WzAyOjAwOmMwOmE4Ojc4OjAxXV0+PC9NQUM+PE5FVFdPUks+PCFbQ0RBVEFbUlRWTEFOM11dPjwvTkVUV09SSz48TkVUV09SS19JR
D48IVtDREFUQVs0XV0+PC9ORVRXT1JLX0lEPjxOSUNfSUQ+PCFbQ0RBVEFbMV1dPjwvTklDX0lEPjxQSFlERVY+PCFbQ0RBVEFbZW
0yXV0+PC9QSFlERVY+PFZMQU4+PCFbQ0RBVEFbWUVTXV0+PC9WTEFOPjxWTEFOX0lEPjwhW0NEQVRBWzNdXT48L1ZMQU5fSUQ+PC9
OSUM+PE9TPjxBUkNIPjwhW0NEQVRBW3g4Nl82NF1dPjwvQVJDSD48Qk9PVD48IVtDREFUQVtoZF1dPjwvQk9PVD48L09TPjxURU1Q
TEFURV9JRD48IVtDREFUQVsxXV0+PC9URU1QTEFURV9JRD48Vk1JRD48IVtDREFUQVsyMTJdXT48L1ZNSUQ+PC9URU1QTEFURT48V
VNFUl9URU1QTEFURT48RkVBVFVSRVM+PEFDUEk+PCFbQ0RBVEFbeWVzXV0+PC9BQ1BJPjwvRkVBVFVSRVM+PEdSQVBISUNTPjxMSV
NURU4+PCFbQ0RBVEFbMC4wLjAuMF1dPjwvTElTVEVOPjxUWVBFPjwhW0NEQVRBW1ZOQ11dPjwvVFlQRT48L0dSQVBISUNTPjwvVVN
FUl9URU1QTEFURT48SElTVE9SWV9SRUNPUkRTLz48L1ZNPg==" ]
CPU="0.2"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6112",
  TYPE="VNC" ]
MEMORY="64"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="1"
VMID="212"
```

Vamos a ver los logs del sistema:

```
[oneadmin@one-node1 one]$ more 212.log
Thu Dec 19 16:48:02 2013 [DiM][I]: New VM state is ACTIVE.
Thu Dec 19 16:48:02 2013 [LCM][I]: New VM state is PROLOG.
Thu Dec 19 16:48:02 2013 [LCM][I]: New VM state is BOOT
Thu Dec 19 16:48:02 2013 [VMM][I]: Generating deployment file: /var/lib/one/vms/212/deployment.0
Thu Dec 19 16:48:02 2013 [VMM][I]: pre: Executed "sudo /usr/sbin/brctl addbr brhm3".
Thu Dec 19 16:48:02 2013 [VMM][I]: pre: Executed "sudo /sbin/ip link set brhm3 up".
Thu Dec 19 16:48:02 2013 [VMM][I]: Device "em2.3" does not exist.
Thu Dec 19 16:48:02 2013 [VMM][I]: pre: Executed "sudo /sbin/vconfig add em2 3".
Thu Dec 19 16:48:02 2013 [VMM][I]: pre: Executed "sudo /sbin/ip link set em2.3 up".
Thu Dec 19 16:48:02 2013 [VMM][I]: pre: Executed "sudo /usr/sbin/brctl addif brhm3 em2.3".
Thu Dec 19 16:48:02 2013 [VMM][I]: ExitCode: 0
Thu Dec 19 16:48:02 2013 [VMM][I]: Successfully execute network driver operation: pre.
Thu Dec 19 16:48:03 2013 [VMM][I]: Command execution fail: cat << EOT | /var/tmp/one/vmm/kvm/deploy
'/var/lib/one//datastores/0/212/deployment.0' 'one-node2' 212 one-node2
Thu Dec 19 16:48:03 2013 [VMM][I]: error: Failed to create domain from
/var/lib/one//datastores/0/212/deployment.0
Thu Dec 19 16:48:03 2013 [VMM][I]: error: Cannot get interface MTU on 'brp0': No such device
Thu Dec 19 16:48:03 2013 [VMM][E]: Could not create domain from
/var/lib/one//datastores/0/212/deployment.0
Thu Dec 19 16:48:03 2013 [VMM][I]: ExitCode: 255
Thu Dec 19 16:48:03 2013 [VMM][I]: Failed to execute virtualization driver operation: deploy.
Thu Dec 19 16:48:03 2013 [VMM][E]: Error deploying virtual machine: Could not create domain from
/var/lib/one//datastores/0/212/deployment.0
Thu Dec 19 16:48:03 2013 [DiM][I]: New VM state is FAILED
```

Bien, el error es que no existe el interfaz bridge brp0, y el sistema no lo crea porque no usamos el driver 802.1Q. Tenemos que preparar y levantar el interfaz brp0 en ambos hosts, y asociarlo al interfaz em2, asegurándonos que queda levantado automáticamente cuando reinicie el máquina. Vamos a configurarlo en ambos hosts. Esto es en el host1:

```
[root@one-node1 network-scripts]# more ifcfg-em2 ifcfg-brp0
::::::::::::::
ifcfg-em2
::::::::::::::
DEVICE="em2"
HWADDR="D4:AE:52:7B:AF:36"
NM_CONTROLLED="no"
ONBOOT="yes"
USERCTL=no
IPV6INIT=no
PEERDNS=no
BRIDGE=brp0
::::::::::::::
ifcfg-brp0
::::::::::::::
DEVICE=brp0
TYPE=Bridge
STP=off
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
```

Vamos a ver si funciona:

```
[root@one-node1 network-scripts]# ifdown em2
bridge brp0 does not exist!
[root@one-node1 network-scripts]# ifup em2
```

```
[root@one-node1 network-scripts]# ifdown em2
[root@one-node1 network-scripts]# ifup em2
[root@one-node1 network-scripts]# brctl show
bridge name     bridge id               STP enabled     interfaces
brp0            8000.d4ae527baf36       no              em2
brvl2           8000.d4ae527baf3a       no              em4.2
```

Pues parece que ha funcionado OK. Ahora hacemos lo mismo con el otro host:

```
[root@one-node2 network-scripts]# more ifcfg-em2 ifcfg-brp0
::::::::::::::
ifcfg-em2
::::::::::::::
DEVICE="em2"
HWADDR="D4:AE:52:7B:AF:9D"
NM_CONTROLLED="no"
ONBOOT="yes"
USERCTL=no
IPV6INIT=no
PEERDNS=no
BRIDGE=brp0
::::::::::::::
ifcfg-brp0
::::::::::::::
DEVICE=brp0
TYPE=Bridge
STP=off
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
[root@one-node2 network-scripts]# ifup em2
[root@one-node2 network-scripts]# brctl show
bridge name     bridge id               STP enabled     interfaces
brhm3           8000.d4ae527baf9d       no              em2.3
brp0            8000.d4ae527baf9d       no              em2
brvl2           8000.d4ae527bafa1       no              em4.2
```

Algo hemos hecho mal en las plantillas, porque utiliza el mismo puerto físico para el interfaz interno y externo del router. Vamos a repasar las plantillas:

```
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 customer1    users        Internet        -               R brp0          0
   3 customer1    users        VLAN3           -               R brhm3         0
   4 customer1    users        RTVLAN3         -               R brhm3         0
[oneadmin@one-node1 mytemplates]$ onevnet show  3
VIRTUAL NETWORK 3 INFORMATION
ID             : 3
```

```
NAME           : VLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brhm3
VLAN           : Yes
PHYSICAL DEVICE: em2
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ onevnet show  4
VIRTUAL NETWORK 4 INFORMATION
ID             : 4
NAME           : RTVLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brhm3
VLAN           : Yes
PHYSICAL DEVICE: em2
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
```

```
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES
```

Vamos a editar las plantillas y actualizarlas:

```
[oneadmin@one-node1 mytemplates]$ more priv_vlan3.net router_vlan3.net
::::::::::::::
priv_vlan3.net
::::::::::::::
NAME = "VLAN3"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brvl3"

NETWORK_ADDRESS = "192.168.120.0/21"
GATEWAY = "192.168.120.1"
DNS = "8.8.8.8 8.8.4.4"
IP_START = "192.168.120.1"
IP_END = "192.168.127.254"
::::::::::::::
router_vlan3.net
::::::::::::::
NAME = "RTVLAN3"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brvl3"

NETWORK_ADDRESS = "192.168.120.0/21"
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER          GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin      oneadmin     VLAN2           -               R brvl2         0
   2 customer1     users        Internet        -               R brp0          0
   3 customer1     users        VLAN3           -               R brhm3         0
   4 customer1     users        RTVLAN3         -               R brhm3         0
```

Como las VLANs están asignadas a los interfaces bridge, no podemos reasignarlas actualizando las plantillas, por lo que tenemos que borrar las plantillas y volver a crearlas:

```
[oneadmin@one-node1 mytemplates]$ onevnet delete 3
[oneadmin@one-node1 mytemplates]$ onevnet delete 4
[oneadmin@one-node1 mytemplates]$ onevnet create priv_vlan3.net
ID: 5
[oneadmin@one-node1 mytemplates]$ onevnet create router_vlan3.net
ID: 6

[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 customer1    users        Internet        -               R brp0          0
   5 oneadmin     oneadmin     VLAN3           -               R brvl3         0
   6 oneadmin     oneadmin     RTVLAN3         -               R brvl3         0
[oneadmin@one-node1 mytemplates]$ onevnet chmod 5 600
[oneadmin@one-node1 mytemplates]$ onevnet chmod 6 600
[oneadmin@one-node1 mytemplates]$ onevnet chown 5 2 1
[oneadmin@one-node1 mytemplates]$ onevnet chown 6 2 1
[oneadmin@one-node1 mytemplates]$ onevnet show 5
VIRTUAL NETWORK 5 INFORMATION
ID             : 5
NAME           : VLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl3
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
```

```
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES

[oneadmin@one-node1 mytemplates]$ onevnet show 6
VIRTUAL NETWORK 6 INFORMATION
ID             : 6
NAME           : RTVLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl3
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES
```

Vamos a probar de nuevo a instanciar el router:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 1 --name "RouterVLAN7" --user customer1
--password 1solomio
VM ID: 213
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   213 customer  users   RouterVLAN7     runn    1     64M one-node2    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm show 213
VIRTUAL MACHINE 213 INFORMATION
ID                  : 213
NAME                : RouterVLAN7
```

```
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node2
START TIME          : 12/19 17:54:01
END TIME            : -
DEPLOY ID           : one-213

VIRTUAL MACHINE MONITORING
NET_TX              : 0K
NET_RX              : 1K
USED CPU            : 1
USED MEMORY         : 64M

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    virtual_router qcow2 non persistent file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet               no brp0         192.101.5.33    02:00:d4:e7:05:21
                                           fe80::400:d4ff:fee7:521
  1 RTVLAN3               yes brvl3        192.168.120.1   02:00:c0:a8:78:01
                                           fe80::400:c0ff:fea8:7801

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START        TIME     PROLOG
  0 one-node2       none             none  12/19 17:54:02   0d 00h00m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DHCP="NO",
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.101.5.62",
  ETH0_IP="192.101.5.33",
  ETH0_MASK="255.255.255.224",
  ETH0_NETWORK="192.101.5.32/27",
  ETH1_IP="192.168.120.1",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="192.168.120.0/21",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",

PRIVNET="PFZORVQ+PElEPjY8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkFN
RT51c2VyczwvR05BTUU+PE5BTUU+UlRWTEFOMzwvTkFNRT48UEVSTUlTU0lPTlM+PE9XTkVSX1U+MTwvT1dORVJfVT48T1dORVJfT
T4xPC9PV05FUl9NPjxPV05FUl9BPjA8L09XTkVSX0E+PEdST1VQX1U+MDwvR1JPVVBfVT48R1JPVVBfTT4wPC9HUk9VUF9NPjxHUk
9VUF9BPjA8L0dST1VQX0E+PE9USEVSX1U+MDwvT1RIRVJfVT48T1RIRVJfTT4wPC9PVEhFUl9NPjxPVEhFUl9BPjA8L09USEVSX0E
+PC9QRVJNSVNTSU9OUz48Q0xVU1RFUl9JRD4tMTwvQ0xVU1RFUl9JRD48Q0xVU1RFUj48L0NMVVNURVI+PFRZUEU+MDwvVFlQRT48
QlJJREdFPmJydmwzPC9CUklER0U+PFZMQU4+MTwvVkxBTj48UEhZREVWPmVtNDwvUEhZREVWPjxWTEFOX0lEPjM8L1ZMQU5fSUQ+P
EdMT0JBTF9QUkVGSVgvPjxTSVRFX1BSRUZJWC8+PFJBTkdFPjxJUF9TVEFSVD4xOTIuMTY4LjEyMC4xPC9JUF9TVEFSVD48SVBfRU
5EPjE5Mi4xNjguMTI3LjI1NDwvSVBfRU5EPjwvUkFOR0U+PFRPVEFMX0xFQVNFUz4xPC9UT1RBTF9MRUFTRVM+PFRFTVBMQVRFPjx
ORVRXT1JLX0FERFJFU1M+PCFbQ0RBVEFbMTkyLjE2OC4xMjAuMC8yMV1dPjwvTkVUV09SS19BRERSRVNTPjxORVRXT1JLX01BU0s+
PCFbQ0RBVEFbMjU1LjI1NS4yNDguMF1dPjwvTkVUV09SS19NQVNLPjwvVEVNUExBVEU+PExFQVNFUz48TEVBU0U+PE1BQz4wMjowM
DpjMDphODo3ODowMTwvTUFDPjxJUD4xOTIuMTY4LjEyMC4xPC9JUD48SVA2X0xJTks+ZmU4MDo6NDAwOmMwZmY6ZmVhODo3ODAxPC
9JUDZfTElOSz48VVNFRD4xPC9VU0VEPjxWSUQ+MjEzPC9WSUQ+PC9MRUFTRT48L0xFQVNFUz48L1ZORVQ+",

PUBNET="PFZORVQ+PElEPjI8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkFNR
T51c2VyczwvR05BTUU+PE5BTUU+SW50ZXJuZXQ8L05BTUU+PFBFUk1JU1NJT05TPjxPV05FUl9VPjE8L09XTkVSX1U+PE9XTkVSX0
0+MTwvT1dORVJfTT48T1dORVJfQT4wPC9PV05FUl9BPjxHUk9VUF9VPjE8L0dST1VQX1U+PEdST1VQX00+MTwvR1JPVVBfTT48R1J
PVVBfQT4wPC9HUk9VUF9BPjxPVEhFUl9VPjE8L09USEVSX1U+PE9USEVSX00+MTwvT1RIRVJfTT48T1RIRVJfQT4wPC9PVEhFUl9B
PjwvUEVSTUlTU0lPTlM+PENMVVNURVJfSUQ+LTE8L0NMVVNURVJfSUQ+PENMVVNURVI+PC9DTFVTVEVSPjxUWVBFPjA8L1RZUEU+P
EJSSURHRT5icnAwPC9CUklER0U+PFZMQU4+MDwvVkxBTj48UEhZREVWPmVtMjwvUEhZREVWPjxWTEFOX0lELz48R0xPQkFMX1BSRU
ZJWC8+PFNJVEVfUFJFRklYLz48UkFOR0U+PElQX1NUQVJUPjIxMi4yMzEuNS4zMzwvSVBfU1RBUlQ+PElQX0VORD4yMTIuMjMxLjU
uNjE8L0lQX0VORD48L1JBTkdFPjxUT1RBTF9MRUFTRVM+MTwvVE9UQUxfTEVBU0VTPjxURU1QTEFURT48RE5TPjwhW0NEQVRBWzgu
OC44LjggOC44LjQuNF1dPjwvRE5TPjxHQVRFV0FZPjwhW0NEQVRBWzIxMi4yMzEuNS42Ml1dPjwvR0FURVdBWT48TkVUV09SS19BR
ERSRVNTPjwhW0NEQVRBWzIxMi4yMzEuNS4zMi8yN11dPjwvTkVUV09SS19BRERSRVNTPjxORVRXT1JLX01BU0s+PCFbQ0RBVEFbMj
U1LjI1NS4yNTUuMjI0XV0+PC9ORVRXT1JLX01BU0s+PC9URU1QTEFURT48TEVBU0VTPjxMRUFTRT48TUFDPjAyOjAwOmQ0OmU3OjA
1OjIxPC9NQUM+PElQPjIxMi4yMzEuNS4zMzwvSVA+PElQNl9MSU5LPmZlODA6OjQwMDpkNGZmOmZlZTc6NTIxPC9JUDZfTElOSz48
VVNFRD4xPC9VU0VEPjxWSUQ+MjEzPC9WSUQ+PC9MRUFTRT48L0xFQVNFUz48L1ZORVQ+",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",

TEMPLATE="PFZNPjxJRD4yMTM8L0lEPjxVSUQ+MjwvVUlEPjxHSUQ+MTwvR0lEPjxVTkFNRT5jdXN0b21lcjE8L1VOQU1FPjxHTkF
NRT51c2VyczwvR05BTUU+PE5BTUU+Um91dGVyVkxBTjc8L05BTUU+PFBFUk1JU1NJT05TPjxPV05FUl9VPjE8L09XTkVSX1U+PE9X
TkVSX00+MTwvT1dORVJfTT48T1dORVJfQT4wPC9PV05FUl9BPjxHUk9VUF9VPjA8L0dST1VQX1U+PEdST1VQX00+MDwvR1JPVVBfT
T48R1JPVVBfQT4wPC9HUk9VUF9BPjxPVEhFUl9VPjA8L09USEVSX1U+PE9USEVSX00+MDwvT1RIRVJfTT48T1RIRVJfQT4wPC9PVE
hFUl9BPjwvUEVSTUlTU0lPTlM+PExBU1RfUE9MTD4wPC9MQVNUX1BPTEw+PFNUQVRFPjE8L1NUQVRFPjxMQ01fU1RBVEU+MDwvTEN
NX1NUQVRFPjxSRVNDSEVEPjA8L1JFU0NIRUQ+PFNUSU1FPjEzODc0NzIwNDE8L1NUSU1FPjxFVElNRT4wPC9FVElNRT48REVQTE9Z
X0lEPjwvREVQTE9ZX0lEPjxNRU1PUlk+MDwvTUVNT1JZPjxDUFU+MDwvQ1BVPjxORVRfVFg+MDwvTkVUX1RYPjxORVRfUlg+MDwvT
kVUX1JYPjxURU1QTEFURT48Q1BVPjwhW0NEQVRBWzAuMl1dPjwvQ1BVPjxESVNLPjxDTE9ORT48IVtDREFUQVtZRVNdXT48L0NMT0
5FPjxEQVRBU1RPUkU+PCFbQ0RBVEFbcWNvdzJfaW1hZ2VzXV0+PC9EQVRBU1RPUkU+PERBVEFTVE9SRV9JRD48IVtDREFUQVsxMDB
dXT48L0RBVEFTVE9SRV9JRD48REVWX1BSRUZJWD48IVtDREFUQVtoZF1dPjwvREVWX1BSRUZJWD48RElTS19JRD48IVtDREFUQVsw
XV0+PC9ESVNLX0lEPjxEUklWRVI+PCFbQ0RBVEFbcWNvdzJdXT48L0RSSVZFUj48SU1BR0U+PCFbQ0RBVEFbdmlydHVhbF9yb3V0Z
XIgcWNvdzIgbm9uIHBlcnNpc3RlbnRdXT48L0lNQUdFPjxJTUFHRV9JRD48IVtDREFUQVs4XV0+PC9JTUFHRV9JRD48UkVBRE9OTF
k+PCFbQ0RBVEFbTk9dXT48L1JFQURPTkxZPjxTQVZFPjwhW0NEQVRBW05PXV0+PC9TQVZFPjxTT1VSQ0U+PCFbQ0RBVEFbL3Zhci9
saWIvb25lL2RhdGFzdG9yZXMvMTAwL2NkZWEzMDIwMjliNzI1ZTc0ZWQyZTY5MTY2ODU4ZDExXV0+PC9TT1VSQ0U+PFRBUkdFVD48
IVtDREFUQVtoZGFdXT48L1RBUkdFVD48VE1fTUFEPjwhW0NEQVRBW3Fjb3cyXV0+PC9UTV9NQUQ+PFRZUEU+PCFbQ0RBVEFbRklMR
V1dPjwvVFlQRT48L0RJU0s+PE1FTU9SWT48IVtDREFUQVs2NF1dPjwvTUVNT1JZPjxOSUM+PEJSSURHRT48IVtDREFUQVticnAwXV
0+PC9CUklER0U+PElQPjwhW0NEQVRBWzIxMi4yMzEuNS4zM11dPjwvSVA+PElQNl9MSU5LPjwhW0NEQVRBW2ZlODA6OjQwMDpkNGZ
```

```
mOmZlZTc6NTIxXV0+PC9JUDZfTElOSz48TUFDPjwhW0NEQVRBWzAyOjAwOmQ0OmU3OjA1OjIxXV0+PC9NQUM+PE5FVFdPUks+PCFb
Q0RBVEFbSW50ZXJuZXRdXT48L05FVFdPUks+PE5FVFdPUktfSUQ+PCFbQ0RBVEFbMl1dPjwvTkVUV09SS19JRD48TklDX0lEPjwhW
0NEQVRBWzBdXT48L05JQ19JRD48UEhZREVWPjwhW0NEQVRBW2VtMl1dPjwvUEhZREVWPjxWTEFOPjwhW0NEQVRBW05PXV0+PC9WTE
FOPjwvTklDPjxOSUM+PEJSSURHRT48IVtDREFUQVticnZsM11dPjwvQlJJREdFPjxJUD48IVtDREFUQVsxOTIuMTY4LjEyMC4xXV0
+PC9JUD48SVA2X0xJTks+PCFbQ0RBVEFbZmU4MDo6NDAwOmMwZmY6ZmVhODo3ODAxXV0+PC9JUDZfTElOSz48TUFDPjwhW0NEQVRB
WzAyOjAwOmMwOmE4Ojc4OjAxXV0+PC9NQUM+PE5FVFdPUks+PCFbQ0RBVEFbUlRWTEFOM11dPjwvTkVUV09SSz48TkVUV09SS19JR
D48IVtDREFUQVs2XV0+PC9ORVRXT1JLX0lEPjxOSUNfSUQ+PCFbQ0RBVEFbMV1dPjwvTklDX0lEPjxQSFlERVY+PCFbQ0RBVEFbZW
00XV0+PC9QSFlERVY+PFZMQU4+PCFbQ0RBVEFbWUVTXV0+PC9WTEFOPjxWTEFOX0lEPjwhW0NEQVRBWzNdXT48L1ZMQU5fSUQ+PC9
OSUM+PE9TPjxBUkNIPjwhW0NEQVRBW3g4Nl82NF1dPjwvQVJDSD48Qk9PVD48IVtDREFUQVtoZF1dPjwvQk9PVD48L09TPjxURU1Q
TEFURV9JRD48IVtDREFUQVsxXV0+PC9URU1QTEFURV9JRD48Vk1JRD48IVtDREFUQVsyMTNdXT48L1ZNSUQ+PC9URU1QTEFURT48V
VNFUl9URU1QTEFURT48RkVBVFVSRVM+PEFDUEk+PCFbQ0RBVEFbeWVzXV0+PC9BQ1BJPjwvRkVBVFVSRVM+PEdSQVBISUNTPjxMSV
NURU4+PCFbQ0RBVEFbMC4wLjAuMF1dPjwvTElTVEVOPjxUWVBFPjwhW0NEQVRBW1ZOQ11dPjwvVFlQRT48L0dSQVBISUNTPjwvVVN
FUl9URU1QTEFURT48SElTVE9SWV9SRUNPUkRTLz48L1ZNPg==" ]

CPU="0.2"

FEATURES=[

  ACPI="yes" ]

GRAPHICS=[

  LISTEN="0.0.0.0",

  PORT="6113",

  TYPE="VNC" ]

MEMORY="64"

OS=[

  ARCH="x86_64",

  BOOT="hd" ]

TEMPLATE_ID="1"

VMID="213"
```

Vamos a migrar la instancia:

```
[oneadmin@one-node1 ~]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   213 customer  users   RouterVLAN7     runn    0   77.9M one-node2    0d 16h54
[oneadmin@one-node1 ~]$ onevm migrate 213 one-node1 --live -v
VM 213: migrating to 0
[oneadmin@one-node1 ~]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   213 customer  users   RouterVLAN7     runn    0   77.9M one-node1    0d 16h55
```

Intentamos hacer un ping, pero no nos responde. Vemos los paquetes ARP llegar correctamente. Ya sabemos lo que pasa: el interfaz brp0 no estaba levantado al levantar el interfaz em2. Con un "ifup brp0" ya se accede a ping sin problemas. Además se puede migrar en vivo la máquina. Ahora vamos a asegurarnos que en la red VLAN3 ninguna máquina de usuario tendrá asignada la IP del default gateway:

```
[oneadmin@one-node1 ~]$ onevnet show 5
VIRTUAL NETWORK 5 INFORMATION
ID             : 5
NAME           : VLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
```

```
BRIDGE         : brvl3
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES

[oneadmin@one-node1 ~]$ onevnet hold 5 "192.168.120.1"
[oneadmin@one-node1 ~]$ onevnet show 5
VIRTUAL NETWORK 5 INFORMATION
ID             : 5
NAME           : VLAN3
USER           : customer1
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl3
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 3
USED LEASES    : 1

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
```

```
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

LEASES ON HOLD
LEASE=[ MAC="02:00:c0:a8:78:01", IP="192.168.120.1", IP6_LINK="fe80::400:c0ff:fea8:7801", USED="1",
VID="-1" ]

VIRTUAL MACHINES

    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME

[oneadmin@one-node1 ~]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          1
   5 customer1    users        VLAN3           -               R brvl3         1
   6 customer1    users        RTVLAN3         -               R brvl3         1
```

Nota: hemos detectado que no se puede invocar instancias si estas usan una red de la que no se es el propietario. Para solucionar el tema de las redes con acceso a internet implantaremos la siguiente política:

- El router virtual será invocado por el administrador. De esta forma se tendrá control de la red pública y se garantizará a cada usuario el acceso a internet mediante al menos una IP pública.
- Cambiamos de propietario la instancia del router para la VLAN3.

Primero cambiamos la red interna correspondiente a la IP utilizada por el router (RTVLAN3):

```
[oneadmin@one-node1 ~]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          0
   5 customer1    users        VLAN3           -               R brvl3         1
   6 customer1    users        RTVLAN3         -               R brvl3         0
[oneadmin@one-node1 ~]$ onevnet chown 6 0 0
[oneadmin@one-node1 ~]$ onevnet show 6
VIRTUAL NETWORK 6 INFORMATION
ID             : 6
NAME           : RTVLAN3
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
```

```
BRIDGE         : brvl3
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 3
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

VIRTUAL MACHINES
```

Ahora hacemos lo mismo con la instancia del router para esa red:

```
[oneadmin@one-node1 ~]$ onetemplate list
  ID USER            GROUP           NAME                                REGTIME
   1 customer1       users           RouterVLAN3                  12/19 16:21:49
   2 oneadmin        oneadmin        tty_linuxV2 non persistent   12/20 11:36:41
[oneadmin@one-node1 ~]$ onetemplate chown 1 0 0
[oneadmin@one-node1 ~]$ onetemplate show 1
TEMPLATE 1 INFORMATION
ID             : 1
NAME           : RouterVLAN3
USER           : oneadmin
GROUP          : oneadmin
REGISTER TIME  : 12/19 16:21:49

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  DHCP="NO",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",
  PRIVNET="$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET="$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",
  TEMPLATE="$TEMPLATE" ]
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="Internet" ]
NIC=[
  IP="192.168.120.1",
  NETWORK="RTVLAN3" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Aprovechamos para aumentar la memoria hasta unos 128M, porque hemos visto que se queda un poco corta.

```
[oneadmin@one-node1 mytemplates]$ onetemplate update 1 router_vlan3.tmpl
[oneadmin@one-node1 mytemplates]$ onetemplate show 1
TEMPLATE 1 INFORMATION
ID              : 1
NAME            : RouterVLAN3
USER            : oneadmin
GROUP           : oneadmin
REGISTER TIME   : 12/19 16:21:49

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
```

```
CONTEXT=[
  DHCP="NO",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",
  PRIVNET="$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET="$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",
  TEMPLATE="$TEMPLATE" ]
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="128"
NAME="RouterVLAN3"
NIC=[
  NETWORK="Internet" ]
NIC=[
  IP="192.168.120.1",
  NETWORK="RTVLAN3" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Ahora probamos a invocar la instancia:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 1 --name "RouterVLAN3"
VM ID: 215
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    3    128M one-node2    0d 00h01
[oneadmin@one-node1 mytemplates]$ onevm show 215
VIRTUAL MACHINE 215 INFORMATION
ID                  : 215
NAME                : RouterVLAN3
USER                : oneadmin
GROUP               : oneadmin
STATE               : ACTIVE
```

```
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node2
START TIME          : 12/22 22:49:36
END TIME            : -
DEPLOY ID           : one-215

VIRTUAL MACHINE MONITORING
NET_TX              : 0K
USED MEMORY         : 128M
NET_RX              : 1K
USED CPU            : 0

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    virtual_router qcow2 non persistent file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet               no brp0         192.101.5.34    02:00:d4:e7:05:22
                                           fe80::400:d4ff:fee7:522
  1 RTVLAN3               yes brvl3        192.168.120.1   02:00:c0:a8:78:01
                                           fe80::400:c0ff:fea8:7801

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node2       none             none  12/22 22:50:02   0d 00h01m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DHCP="NO",
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.101.5.62",
  ETH0_IP="192.101.5.34",
  ETH0_MASK="255.255.255.224",
  ETH0_NETWORK="192.101.5.32/27",
  ETH1_IP="192.168.120.1",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="192.168.120.0/21",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",

PRIVNET="PFZORVQ+PElEPjY8L0lEPjxVSUQ+MDwvVUlEPjxHSUQ+MDwvR0lEPjxVTkFNRT5vbmVhZG1pbjwvVU5BTUU+PEdOQU1F
Pm9uZWFkbWluPC9HTkFNRT48TkFNRT5SVFZMQU4zPC9OQU1FPjxQRVJNSVNTSU9OUz48T1dORVJfVT4xPC9PV05FUl9VPjxPV05FU
l9NPjE8L09XTkVSX00+PE9XTkVSX0E+MDwvT1dORVJfQT48R1JPVVBfVT4wPC9HUk9VUF9VPjxHUk9VUF9NPjA8L0dST1VQX00+PE
dST1VQX0E+MDwvR1JPVVBfQT48T1RIRVJfVT4wPC9PVEhFUl9VPjxPVEhFUl9NPjA8L09USEVSX00+PE9USEVSX0E+MDwvT1RIRVJ
fQT48L1BFUk1JU1NJT05TPjxDTFVTVEVSX0lEPi0xPC9DTFVTVEVSX0lEPjxDTFVTVEVSPjwvQ0xVU1RFUj48VFlQRT4wPC9UWVBF
PjxCUklER0U+YnJ2bDM8L0JSSURHRT48VkxBTj4xPC9WTEFOPjxQSFlERVY+ZW00PC9QSFlERVY+PFZMQU5fSUQ+MzwvVkxBTl9JR
D48R0xPQkFMX1BSRUZJWC8+PFNJVEVfUFJFRklYLz48UkFOR0U+PElQX1NUQVJUPjE5Mi4xNjguMTIwLjE8L0lQX1NUQVJUPjxJUF
9FTkQ+MTkyLjE2OC4xMjcuMjU0PC9JUF9FTkQ+PC9SQU5HRT48VE9UQUxfTEVBU0VTPjE8L1RPVEFMX0xFQVNFUz48VEVNUExBVEU
+PE5FVFdPUktfQUREUkVTUz48IVtDREFUQVsxOTIuMTY4LjEyMC4wLzIxXV0+PC9ORVRXT1JLX0FERFJFU1M+PE5FVFdPUktfTUFT
Sz48IVtDREFUQVsyNTUuMjU1LjI0OC4wXV0+PC9ORVRXT1JLX01BU0s+PC9URU1QTEFURT48TEVBU0VTPjxMRUFTRT48TUFDPjAyO
jAwOmMwOmE4Ojc4OjAxPC9NQUM+PElQPjE5Mi4xNjguMTIwLjE8L0lQPjxJUDZfTElOSz5mZTgwOjo0MDA6YzBmZjpmZWE4Ojc4MD
E8L0lQNl9MSU5LPjxVU0VEPjE8L1VTRUQ+PFZJRD4yMTU8L1ZJRD48L0xFQVNFPjwvTEVBU0VTPjwvVk5FVD4=",

PUBNET="PFZORVQ+PElEPjI8L0lEPjxVSUQ+MDwvVUlEPjxHSUQ+MDwvR0lEPjxVTkFNRT5vbmVhZG1pbjwvVU5BTUU+PEdOQU1FP
m9uZWFkbWluPC9HTkFNRT48TkFNRT5JbnRlcm5ldDwvTkFNRT48UEVSTUlTU0lPTlM+PE9XTkVSX1U+MTwvT1dORVJfVT48T1dORV
JfTT4xPC9PV05FUl9NPjxPV05FUl9BPjA8L09XTkVSX0E+PEdST1VQX1U+MTwvR1JPVVBfVT48R1JPVVBfTT4xPC9HUk9VUF9NPjx
HUk9VUF9BPjA8L0dST1VQX0E+PE9USEVSX1U+MTwvT1RIRVJfVT48T1RIRVJfTT4xPC9PVEhFUl9NPjxPVEhFUl9BPjA8L09USEVS
X0E+PC9QRVJNSVNTSU9OUz48Q0xVU1RFUl9JRD4tMTwvQ0xVU1RFUl9JRD48Q0xVU1RFUj48L0NMVVNURVI+PFRZUEU+MDwvVFlQR
T48QlJJREdFPmJycDA8L0JSSURHRT48VkxBTj4wPC9WTEFOPjxQSFlERVY+ZW0yPC9QSFlERVY+PFZMQU5fSUQvPjxHTE9CQUxfUF
JFRklYLz48U0lURV9QUkVGSVgvPjxSQU5HRT48SVBfU1RBUlQ+MjEyLjIzMS41LjMzPC9JUF9TVEFSVD48SVBfRU5EPjIxMi4yMzE
uNS42MTwvSVBfRU5EPjwvUkFOR0U+PFRPVEFMX0xFQVNFUz4xPC9UT1RBTF9MRUFTRVM+PFRFTVBMQVRFPjxETlM+PCFbQ0RBVEFb
OC44LjguOCA4LjguNC40XV0+PC9ETlM+PEdBVEVXQVk+PCFbQ0RBVEFbMjEyLjIzMS41LjYyXV0+PC9HQVRFV0FZPjxORVRXT1JLX
0FERFJFU1M+PCFbQ0RBVEFbMjEyLjIzMS41LjMyLzI3XV0+PC9ORVRXT1JLX0FERFJFU1M+PE5FVFdPUktfTUFTSz48IVtDREFUQV
syNTUuMjU1LjI1NS4yMjRdXT48L05FVFdPUktfTUFTSz48L1RFTVBMQVRFPjxMRUFTRVM+PExFQVNFPjxNQUM+MDI6MDA6ZDQ6ZTc
6MDU6MjI8L01BQz48SVA+MjEyLjIzMS41LjM0PC9JUD48SVA2X0xJTks+ZmU4MDo6NDAwOmQ0ZmY6ZmVlNzo1MjI8L0lQNl9MSU5L
PjxVU0VEPjE8L1VTRUQ+PFZJRD4yMTU8L1ZJRD48L0xFQVNFPjwvTEVBU0VTPjwvVk5FVD4=",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",

TEMPLATE="PFZNPjxJRD4yMTU8L0lEPjxVSUQ+MDwvVUlEPjxHSUQ+MDwvR0lEPjxVTkFNRT5vbmVhZG1pbjwvVU5BTUU+PEdOQU1
FPm9uZWFkbWluPC9HTkFNRT48TkFNRT5Sb3V0ZXJWTEFOMzwvTkFNRT48UEVSTUlTU0lPTlM+PE9XTkVSX1U+MTwvT1dORVJfVT48
T1dORVJfTT4xPC9PV05FUl9NPjxPV05FUl9BPjA8L09XTkVSX0E+PEdST1VQX1U+MDwvR1JPVVBfVT48R1JPVVBfTT4wPC9HUk9VU
F9NPjxHUk9VUF9BPjA8L0dST1VQX0E+PE9USEVSX1U+MDwvT1RIRVJfVT48T1RIRVJfTT4wPC9PVEhFUl9NPjxPVEhFUl9BPjA8L0
9USEVSX0E+PC9QRVJNSVNTSU9OUz48TEFTVF9QT0xMPjA8L0xBU1RfUE9MTD48U1RBVEU+MTwvU1RBVEU+PExDTV9TVEFURT4wPC9
MQ01fU1RBVEU+PFJFU0NIRUQ+MDwvUkVTQ0hFRD48U1RJTUU+MTM4Nzc0ODk3NjwvU1RJTUU+PEVUSU1FPjA8L0VUSU1FPjxERVBM
T1lfSUQ+PC9ERVBMT1lfSUQ+PE1FTU9SWT4wPC9NRU1PUlk+PENQVT4wPC9DUFU+PE5FVF9UWD4wPC9ORVRfVFg+PE5FVF9SWD4wP
C9ORVRfUlg+PFRFTVBMQVRFPjxDUFU+PCFbQ0RBVEFbMC4yXV0+PC9DUFU+PERJU0s+PENMT05FPjwhW0NEQVRBW1llFU11dPjwvQ0
xPTkU+PERBVEFTVE9SRT48IVtDREFUQVtxY293Ml9pbWFnZXNdXT48L0RBVEFTVE9SRT48REFUQVNUT1JFX0lEPjwhW0NEQVRBWzE
wMF1dPjwvREFUQVNUT1JFX0lEPjxERVZfUFJFRklYPjwhW0NEQVRBW2hkXV0+PC9ERVZfUFJFRklYPjxESVNLX0lEPjwhW0NEQVRB
WzBdXT48L0RJU0tfSUQ+PERSSVZFUj48IVtDREFUQVtxY293Ml1dPjwvRFJJVkVSPjxJTUFHRT48IVtDREFUQVt2aXJ0dWFsX3Jvd
XRlciBxY293MiBub24gcGVyc2lzdGVudF1dPjwvSU1BR0U+PElNQUdFX0lEPjwhW0NEQVRBWzhdXT48L0lNQUdFX0lEPjxSRUFET0
5MWT48IVtDREFUQVtOT11dPjwvUkVBRE9OTFk+PFNBVkU+PCFbQ0RBVEFbTk9dXT48L1NBVkU+PFNPVVJDRT48IVtDREFUQVsvdmF
yL2xpYi9vbmUvZGF0YXN0b3Jlcy8xMDAvY2RlYTMwMjAyOWI3MjVlNzRlZDJlNjkxNjY4NThkMTFdXT48L1NPVVJDRT48VEFSR0VU
PjwhW0NEQVRBW2hkYV1dPjwvVEFSR0VUPjxUTV9NQUQ+PCFbQ0RBVEFbcWNvdzJdXT48L1RNX01BRD48VFlQRT48IVtDREFUQVtGS
UxFXV0+PC9UWVBFPjwvRElTSz48TUVNT1JZPjwhW0NEQVRBWzEyOF1dPjwvTUVNT1JZPjxOSUM+PEJSSURHRT48IVtDREFUQVticn
AwXV0+PC9CUklER0U+PElQPjwhW0NEQVRBWzIxMi4yMzEuNS4zNF1dPjwvSVA+PElQNl9MSU5LPjwhW0NEQVRBW2ZlODA6OjQwMDp
kNGZmOmZlZTc6NTIyXV0+PC9JUDZfTElOSz48TUFDPjwhW0NEQVRBWzAyOjAwOmQ0OmU3OjA1OjIyXV0+PC9NQUM+PE5FVFdPUks+
PCFbQ0RBVEFbSW50ZXJuZXRdXT48L05FVFdPUks+PE5FVFdPUktfSUQ+PCFbQ0RBVEFbMl1dPjwvTkVUV09SS19JRD48TklDX0lEP
jwhW0NEQVRBWzBdXT48L05JQ19JRD48UEhZREVWPjwhW0NEQVRBW2VtMl1dPjwvUEhZREVWPjxWTEFOPjwhW0NEQVRBW05PXV0+PC
9WTEFOPjwvTklDPjxOSUM+PEJSSURHRT48IVtDREFUQVticnZsZsM11dPjwvQlJJREdFPjxJUD48IVtDREFUQVsxOTIuMTY4LjEyMC4
```

```
xXV0+PC9JUD48SVA2X0xJTks+PCFbQ0RBVEFbZmU4MDo6NDAwOmMwZmY6ZmVhODo3ODAxXV0+PC9JUDZfTElOSz48TUFDPjwhW0NE
QVRBWzAyOjAwOmMwOmE4Ojc4OjAxXV0+PC9NQUM+PE5FVFdPUks+PCFbQ0RBVEFbUlRWTEFOM11dPjwvTkVUV09SSz48TkVUV09SS
19JRD48IVtDREFUQVs2XV0+PC9ORVRXT1JLX0lEPjxOSUNfSUQ+PCFbQ0RBVEFbMV1dPjwvTklDX0lEPjxQSFlERVY+PCFbQ0RBVE
FbZW00XV0+PC9QSFlERVY+PFZMQU4+PCFbQ0RBVEFbWUVTXV0+PC9WTEFOPjxWTEFOX0lEPjwhW0NEQVRBWzNdXT48L1ZMQU5fSUQ
+PC9OSUM+PE9TPjxBUkNIPjwhW0NEQVRBW3g4Nl82NF1dPjwvQVJDSD48Qk9PVD48IVtDREFUQVtoZF1dPjwvQk9PVD48L09TPjxU
RU1QTEFURV9JRD48IVtDREFUQVsxXV0+PC9URU1QTEFURV9JRD48Vk1JRD48IVtDREFUQVsyMTVdXT48L1ZNSUQ+PC9URU1QTEFUR
T48VVNFUl9URU1QTEFURT48RkVBVFVSRVM+PEFDUEk+PCFbQ0RBVEFbeWVzXV0+PC9BQ1BJPjwvRkVBVFVSRVM+PEdSQVBISUNTPj
xMSVNURU4+PCFbQ0RBVEFbMC4wLjAuMF1dPjwvTElTVEVOPjxUWVBFPjwhW0NEQVRBW1ZOQ11dPjwvVFlQRT48L0dSQVBISUNTPjw
vVVNFUl9URU1QTEFURT48SElTVE9SWV9SRUNPUkRTLz48L1ZNPg==" ]
```

```
CPU="0.2"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6115",
  TYPE="VNC" ]
MEMORY="128"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="1"
VMID="215"
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -              R brvl2         0
   2 oneadmin     oneadmin     Internet        -              R brp0          1
   5 customer1    users        VLAN3           -              R brvl3         1
   6 oneadmin     oneadmin     RTVLAN3         -              R brvl3         1
```

Por el momento todo funciona OK. Ahora nos toca crear una plantilla que pueda utilizar el usuario customer1 para invocar máquinas en la VLAN3.

```
[oneadmin@one-node1 mytemplates]$ more ttyVL3_contextNP.tmpl
NAME="tty_linuxV2 VLAN3 NP"
CPU="0.1"
DISK=[
  IMAGE_ID="6" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
FEATURES=[
  ACPI="no" ]
NIC = [ NETWORK="VLAN3" ]

CONTEXT=[
  NETWORK        = "YES" ]
[oneadmin@one-node1 mytemplates]$ onetemplate create ttyVL3_contextNP.tmpl
ID: 3
```

```
[oneadmin@one-node1 mytemplates]$ onetemplate chown 3 2 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 3
TEMPLATE 3 INFORMATION
ID             : 3
NAME           : tty_linuxV2 VLAN3 NP
USER           : customer1
GROUP          : users
REGISTER TIME  : 12/22 23:15:56

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES" ]
CPU="0.1"
DISK=[
  IMAGE_ID="6" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="VLAN3" ]
```

Con esto el customer1 ya tiene una plantilla con la que poder invocar máquinas virtuales en la VLAN3 con salida a internet. Vamos a invocarle una instancia con su usuario y password:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 3 --name "tty_vlan3_1" --user customer1
--password 1solomio
VM ID: 216
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 00h30
   216 customer users    tty_vlan3_1     boot    0      0K one-node1    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 00h30
   216 customer users    tty_vlan3_1     runn   94     64M one-node1    0d 00h00
```

Vemos que la instancia se ha creado correctamente:

```
[oneadmin@one-node1 mytemplates]$ onevm show 216
VIRTUAL MACHINE 216 INFORMATION
ID                  : 216
```

```
NAME                : tty_vlan3_1
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node1
START TIME          : 12/22 23:19:53
END TIME            : -
DEPLOY ID           : one-216

VIRTUAL MACHINE MONITORING
USED CPU            : 5
USED MEMORY         : 64M
NET_TX              : 0K
NET_RX              : 0K

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    tty_linuxV2 qcow2 non persistent    file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.120.2   02:00:c0:a8:78:02
                                           fe80::400:c0ff:fea8:7802

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node1       none             none   12/22 23:20:02    0d 00h01m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.120.2",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  NETWORK="YES",
  TARGET="hdb" ]
```

```
CPU="0.1"
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6116",
  TYPE="VNC" ]
MEMORY="64"
TEMPLATE_ID="3"
VMID="216"
```

Vamos a verlo... Se nos ha corrompido la imagen del tty_linux que habíamos contextualizado.

4.3.2. Recuperando la imagen tty_linux

Vamos a probar con la imagen raw original, a ver si conseguimos que arranque:

```
[oneadmin@one-node1 mytemplates]$ oneimage show 7
IMAGE 7 INFORMATION
ID             : 7
NAME           : tty_linuxV2 raw non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 12/19 12:19:47
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/ca5a240f355ce8b4fed33f77946f47dc
PATH           : /home/datastores/tty_linuxV2.img
SIZE           : 40M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV2 contextualized with raw driver non persistent"
DEV_PREFIX="hd"
DRIVER="raw"

VIRTUAL MACHINES
```

Modificamos la plantilla anterior para usar la imagen raw, a ver si tenemos suerte:

```
[oneadmin@one-node1 mytemplates]$ onetemplate update 3 ttyVL3_contextNP.tmpl
```

```
[oneadmin@one-node1 mytemplates]$ onetemplate show 3
TEMPLATE 3 INFORMATION
ID             : 3
NAME           : tty_linuxV2 VLAN3 NP
USER           : customer1
GROUP          : users
REGISTER TIME  : 12/22 23:15:56

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES" ]
CPU="0.1"
DISK=[
  IMAGE_ID="7" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NAME="tty_linuxV2 VLAN3 NP"
NIC=[
  NETWORK="VLAN3" ]
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 3 --name "tty_vlan3_1" --user customer1
--password 1solomio
VM ID: 217
```

Le ocurre exactamente lo mismo que en el caso anterior, pero hemos conseguido reparar la imagen, y al arrancar de nuevo funciona correctamente. Como la imagen raíz está en mal estado, tenemos que salvar la imagen actual con una copia (ya que no es una imagen persistente), y usarla como imagen matriz para remplazar a las dos anteriores. Vamos a ello. Para trabajar de forma mas eficiente, accederemos a la máquina por ssh a través del port forwarding realizado por el router de la VLAN3. Ahora mismo tenemos esto:

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 01h18
   217 customer users    tty_vlan3_1     runn    5     64M one-node1    0d 00h06
[oneadmin@one-node1 mytemplates]$ onevm show 217
VIRTUAL MACHINE 217 INFORMATION
ID                  : 217
NAME                : tty_vlan3_1
```

```
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node1
START TIME          : 12/23 00:01:30
END TIME            : -
DEPLOY ID           : one-217

VIRTUAL MACHINE MONITORING
NET_RX              : 0K
USED MEMORY         : 64M
NET_TX              : 0K
USED CPU            : 5

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    tty_linuxV2 raw non persistent      file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.120.2   02:00:c0:a8:78:02
                                           fe80::400:c0ff:fea8:7802

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node1       none             none  12/23 00:01:32   0d 00h08m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.120.2",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  NETWORK="YES",
  TARGET="hdb" ]
CPU="0.1"
```

```
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6117",
  TYPE="VNC" ]
MEMORY="64"
TEMPLATE_ID="3"
VMID="217"
```

Vemos que la instancia del router y la de la máquina virtual están en host diferentes, pero eso no debe de afectar en nada ya que la VLAN3 existe en ambos hosts y se encuentra interconectada a través del interfaz de red em4.3. Vamos a acceder desde mi equipo local, atravesando el virtual router y la VLAN3 hasta el host2:

```
[jamontes@Tesla ~]$ ssh root@192.101.5.34 -p 2222
The authenticity of host '[192.101.5.34]:2222 ([192.101.5.34]:2222)' can't be established.
RSA key fingerprint is 5b:d6:3a:a9:8a:53:21:66:70:0c:b7:26:34:45:b1:27.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.101.5.34]:2222' (RSA) to the list of known hosts.
root@192.101.5.34's password:

Chop wood, carry water.

[root@ttylinux_host ~]#
```

Estupendo. Todo funciona a la primera.

```
[root@ttylinux_host ~]# who
USER       TTY      IDLE      TIME            HOST
root       tty1     00:08     Dec 22 23:04:10
root       pts/0    00:00     Dec 22 23:12:29 83.194.241.151
```

Hacemos un par de pruebas de reinicio para asegurarnos que la imagen está en perfecto estado, y que podremos usarla sin problemas tan pronto como hayamos hecho una copia. Una vez parada, con el comando "shutdown -h" es el momento de salvar la imagen:

```
[oneadmin@one-node1 mytemplates]$ onevm disk-snapshot tty_vlan3_1 0 "tty_linuxV3 raw"
Image ID: 11
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default        11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default        10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
   6 oneadmin   oneadmin   tty_linuxV2 qco qcow2_imag     11M OS    No rdy     0
   7 oneadmin   oneadmin   tty_linuxV2 raw default        40M OS    No used    1
   8 oneadmin   oneadmin   virtual_router  qcow2_imag    191M OS    No used    1
   9 oneadmin   oneadmin   Ubuntu Server 1 qcow2_imag    589M OS    No rdy     0
  10 oneadmin   oneadmin   CentOS 6.2 non  qcow2_imag    1.8G OS    No rdy     0
  11 oneadmin   oneadmin   tty_linuxV3 raw default         40M OS    No lock    0
```

Vemos que se ha quedado en estado lock. Esto lo hace para preservar la imagen, hasta que apaguemos la maquina mediante el API. Como esta máquina virtual no soporta ACPI, tendremos que apagarla en modo "hard":

```
[oneadmin@one-node1 mytemplates]$ onevm shutdown tty_vlan3_1 --hard
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 01h52
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default         40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default         83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default         11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default         10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
   6 oneadmin   oneadmin   tty_linuxV2 qco qcow2_imag     11M OS    No rdy     0
   7 oneadmin   oneadmin   tty_linuxV2 raw default         40M OS    No rdy     0
   8 oneadmin   oneadmin   virtual_router  qcow2_imag    191M OS    No used    1
   9 oneadmin   oneadmin   Ubuntu Server 1 qcow2_imag    589M OS    No rdy     0
  10 oneadmin   oneadmin   CentOS 6.2 non  qcow2_imag    1.8G OS    No rdy     0
  11 oneadmin   oneadmin   tty_linuxV3 raw default         40M OS    No rdy     0
[oneadmin@one-node1 mytemplates]$ oneimage show 11
IMAGE 11 INFORMATION
ID             : 11
NAME           : tty_linuxV3 raw
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 12/23 00:36:15
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/1/dc3237e3b8b813c200248e0598a834c4
FSTYPE         : save_as
SIZE           : 40M
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DEV_PREFIX="hd"
```

```
DRIVER="raw"
SAVED_DISK_ID="0"
SAVED_IMAGE_ID="7"
SAVED_VM_ID="218"
SAVE_AS="YES"

VIRTUAL MACHINES
```

Bien, de momento hemos salvado la imagen raw. Vamos a probar de nuevo actualizando la plantilla:

```
[oneadmin@one-node1 mytemplates]$ onetemplate update 3 ttyVL3_contextNP.tmpl
[oneadmin@one-node1 mytemplates]$ onetemplate show 3
TEMPLATE 3 INFORMATION
ID             : 3
NAME           : tty_linuxV2 VLAN3 NP
USER           : customer1
GROUP          : users
REGISTER TIME  : 12/22 23:15:56

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES" ]
CPU="0.1"
DISK=[
  IMAGE_ID="11" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NAME="tty_linuxV2 VLAN3 NP"
NIC=[
  NETWORK="VLAN3" ]
```

Volvemos a instanciar con la nueva imagen:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 3 --name "tty_vlan3_1" --user customer1
--password 1solomio
[TemplateInstantiate] User [2] : Not authorized to perform USE IMAGE [11].
```

Cierto; se nos ha olvidado dar permisos de uso a la imagen:

```
[oneadmin@one-node1 mytemplates]$ oneimage chmod 11 644
```

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 3 --name "tty_vlan3_1" --user customer1
--password 1solomio
VM ID: 219
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 01h59
   219 customer users    tty_vlan3_1     runn   26     64M one-node1    0d 00h00
```

Bien, esta vez sí que funciona correctamente, con la imagen tipo raw. Eliminamos la instancia y exportamos la imagen tipo raw en formato qcow2, para volver a importarla en el datastore correctamente.

```
[oneadmin@one-node1 mytemplates]$ onevm delete 219
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 02h01
[root@one-node1 ~]# cd /home/datastores/
[root@one-node1 datastores]# qemu-img convert -O qcow2
/var/lib/one/datastores/1/dc3237e3b8b813c200248e0598a834c4 tty_linuxV3.qcow2
[root@one-node1 datastores]# qemu-img info tty_linuxV3.qcow2
image: tty_linuxV3.qcow2
file format: qcow2
virtual size: 40M (41943040 bytes)
disk size: 10M
cluster_size: 65536
```

Ahora ya podemos crear la plantilla para importar la imagen en el datastore:

```
[oneadmin@one-node1 mytemplates]$ more tty_linuxV3_qcow2.tmpl
NAME          = "tty_linuxV3 qcow2 non persistent"
PATH          = /home/datastores/tty_linuxV3.qcow2
TYPE          = OS
DRIVER        = qcow2
DESCRIPTION   = "tty_linuxV3 contextualized with qcow2 driver non persistent"
[oneadmin@one-node1 mytemplates]$ oneimage create -d qcow2_images tty_linuxV3_qcow2.tmpl
ID: 12
[oneadmin@one-node1 mytemplates]$ oneimage chmod 12 644
[oneadmin@one-node1 mytemplates]$ oneimage show 12
IMAGE 12 INFORMATION
ID             : 12
NAME           : tty_linuxV3 qcow2 non persistent
USER           : oneadmin
GROUP          : oneadmin
DATASTORE      : qcow2_images
TYPE           : OS
REGISTER TIME  : 12/23 00:59:54
PERSISTENT     : No
SOURCE         : /var/lib/one/datastores/100/a39b7191c665bd2f57129a70b63838b3
PATH           : /home/datastores/tty_linuxV3.qcow2
SIZE           : 11M
```

```
STATE           : rdy
RUNNING_VMS     : 0

PERMISSIONS
OWNER           : um-
GROUP           : u--
OTHER           : u--

IMAGE TEMPLATE
DESCRIPTION="tty_linuxV3 contextualized with qcow2 driver non persistent"
DEV_PREFIX="hd"
DRIVER="qcow2"

VIRTUAL MACHINES
```

Ahora volvemos a actualizar la plantilla para usar esta imagen en lugar de la imagen raw:

```
[oneadmin@one-node1 mytemplates]$ onetemplate update 3 ttyVL3_contextNP.tmpl
[oneadmin@one-node1 mytemplates]$ onetemplate show 3
TEMPLATE 3 INFORMATION
ID              : 3
NAME            : tty_linuxV2 VLAN3 NP
USER            : customer1
GROUP           : users
REGISTER TIME   : 12/22 23:15:56

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES" ]
CPU="0.1"
DISK=[
  IMAGE_ID="12" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NAME="tty_linuxV2 VLAN3 NP"
NIC=[
  NETWORK="VLAN3" ]
```

E instanciamos la máquina con la imagen qcow2:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 3 --name "tty_vlan3_1" --user customer1
--password 1solomio
VM ID: 220
[oneadmin@one-node1 mytemplates]$ onevm show 220
VIRTUAL MACHINE 220 INFORMATION
ID                  : 220
NAME                : tty_vlan3_1
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node1
START TIME          : 12/23 01:03:20
END TIME            : -
DEPLOY ID           : one-220

VIRTUAL MACHINE MONITORING
NET_TX              : 0K
USED MEMORY         : 64M
USED CPU            : 5
NET_RX              : 0K

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                              TYPE SAVE SAVE_AS
  0 hda    tty_linuxV3 qcow2 non persistent    file   NO       -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.120.2   02:00:c0:a8:78:02
                                           fe80::400:c0ff:fea8:7802

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node1       none             none  12/23 01:03:32    0d 00h00m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.120.2",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  NETWORK="YES",
  TARGET="hdb" ]
CPU="0.1"
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6120",
  TYPE="VNC" ]
MEMORY="64"
TEMPLATE_ID="3"
VMID="220"
```

Vamos a probar el acceso por ssh:

```
[jamontes@Tesla ~]$ ssh root@192.101.5.34 -p 2222
root@192.101.5.34's password:

Chop wood, carry water.

[root@ttylinux_host ~]# who
USER       TTY      IDLE      TIME            HOST
root       pts/0    00:00     Dec 23 00:05:08 83.194.241.151
```

Bien. Ahora tenemos que hacer limpieza de imágenes, eliminando aquellas que sabemos están corruptas:

```
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default        40M OS    No rdy     0
   1 oneadmin   oneadmin   OpenNebula 4.2  default        83M OS    No rdy     0
   3 oneadmin   oneadmin   Ubuntu Server 1 default        11G OS    No rdy     0
   4 oneadmin   oneadmin   CentOS Server 6 default        10G OS    No rdy     0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy     0
   6 oneadmin   oneadmin   tty_linuxV2 qco qcow2_imag     11M OS    No rdy     0
   7 oneadmin   oneadmin   tty_linuxV2 raw default        40M OS    No rdy     0
   8 oneadmin   oneadmin   virtual_router  qcow2_imag    191M OS    No used    1
   9 oneadmin   oneadmin   Ubuntu Server 1 qcow2_imag    589M OS    No rdy     0
  10 oneadmin   oneadmin   CentOS 6.2 non  qcow2_imag    1.8G OS    No rdy     0
  11 oneadmin   oneadmin   tty_linuxV3 raw default        40M OS    No rdy     0
  12 oneadmin   oneadmin   tty_linuxV3 qco qcow2_imag     11M OS    No used    1
```

Tenemos que eliminar las dos imágenes tty_linuxV2:

```
[oneadmin@one-node1 mytemplates]$ oneimage delete 6
[oneadmin@one-node1 mytemplates]$ oneimage delete 7
```

```
[oneadmin@one-node1 mytemplates]$ oneimage list
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT RVMS
   0 oneadmin   oneadmin   ttylinux        default         40M OS    No rdy      0
   1 oneadmin   oneadmin   OpenNebula 4.2  default         83M OS    No rdy      0
   3 oneadmin   oneadmin   Ubuntu Server 1 default         11G OS    No rdy      0
   4 oneadmin   oneadmin   CentOS Server 6 default         10G OS    No rdy      0
   5 oneadmin   oneadmin   CentOS 6.5 non  qcow2_imag    3.2G OS    No rdy      0
   8 oneadmin   oneadmin   virtual_router  qcow2_imag    191M OS    No used     1
   9 oneadmin   oneadmin   Ubuntu Server 1 qcow2_imag    589M OS    No rdy      0
  10 oneadmin   oneadmin   CentOS 6.2 non  qcow2_imag    1.8G OS    No rdy      0
  11 oneadmin   oneadmin   tty_linuxV3 raw default         40M OS    No rdy      0
  12 oneadmin   oneadmin   tty_linuxV3 qco qcow2_imag     11M OS    No used     1
```

4.3.3. Preparando plantillas de prueba para customer1

Ahora vamos a preparar algunas plantillas de prueba al usuario customer1:

```
[oneadmin@one-node1 mytemplates]$ more centos65VL3_contextNP.tmpl
NAME="Centos65_x86_64 VLAN3 NP"
CPU="0.5"
DISK=[
  IMAGE_ID="5" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
FEATURES=[
  ACPI="yes" ]
NIC = [ NETWORK="VLAN3" ]

CONTEXT=[
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
```

Para que el usuario pueda acceder a la máquina es necesario que en su perfil de usuario introduzca a clave SSH pública (a través de la web).

```
[oneadmin@one-node1 mytemplates]$ onetemplate create centos65VL3_contextNP.tmpl
ID: 4
[oneadmin@one-node1 mytemplates]$ onetemplate chown 4 2 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 4
TEMPLATE 4 INFORMATION
ID              : 4
NAME            : Centos65_x86_64 VLAN3 NP
USER            : customer1
GROUP           : users
REGISTER TIME   : 12/23 01:35:05
```

```
PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.5"
DISK=[
  IMAGE_ID="5" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
NIC=[
  NETWORK="VLAN3" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Instanciamos una máquina para comprobar que la plantilla y la imagen funcionan OK. Primero eliminamos la instancia que tenemos corriendo, para que la nueva instancia tome la IP 2 y podamos acceder por ssh:

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 02h48
   220 customer users    tty_vlan3_1     runn    5     64M one-node1    0d 00h34
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 02h50
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 4 --name "centro65_vlan3_1" --user
customer1 --password 1solomio
VM ID: 221
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 02h52
   221 customer users    centro65_vlan3_ runn    0      0K one-node1    0d 00h00
```

Vamos a acceder por ssh desde afuera. Primero tenemos que eliminar la entrada en el known_hosts de mi equipo local.

```
[jamontes@Tesla .ssh]$ ssh root@192.101.5.34 -p 2222
The authenticity of host '[192.101.5.34]:2222 ([192.101.5.34]:2222)' can't be established.
```

```
RSA key fingerprint is 3f:d6:b0:75:21:0a:3e:93:53:5a:ee:8e:b9:8a:9e:17.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.101.5.34]:2222' (RSA) to the list of known hosts.
root@192.101.5.34's password:
Last login: Thu Dec  5 13:03:35 2013
[root@localhost ~]# uname -a
Linux localhost.localdomain 2.6.32-431.el6.x86_64 #1 SMP Fri Nov 22 03:15:09 UTC 2013 x86_64 x86_64
x86_64 GNU/Linux
[root@localhost ~]# cat /etc/redhat-release
CentOS release 6.5 (Final)
[root@localhost ~]# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:C0:A8:78:02
          inet addr:192.168.120.2  Bcast:192.168.127.255  Mask:255.255.248.0
          inet6 addr: fe80::c0ff:fea8:7802/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:153 errors:0 dropped:0 overruns:0 frame:0
          TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15867 (15.4 KiB)  TX bytes:16392 (16.0 KiB)
          Interrupt:10 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:200 (200.0 b)  TX bytes:200 (200.0 b)
```

Esto ha funcionado bien.

4.3.4. Trabajando con la persistencia de las imágenes

Para demostrar la persistencia de la imagen, mientras no la apaguemos del todo (con los comandos shutdown o delete del API), vamos a reiniciarla:

```
[root@localhost ~]# reboot

Broadcast message from root@localhost.localdomain
        (/dev/pts/0) at 1:48 ...

The system is going down for reboot NOW!
[root@localhost ~]# Connection to 192.101.5.34 closed by remote host.
Connection to 192.101.5.34 closed.
```

```
[jamontes@Tesla .ssh]$ ssh root@192.101.5.34 -p 2222
root@192.101.5.34's password:
Last login: Mon Dec 23 01:45:19 2013 from adijon-151-1-4-151.w83-194.abo.wanadoo.fr
[root@localhost ~]#
```

Vemos que ha reiniciado correctamente, y como tiene la resolución por DNS, nos dice que la última entrada al sistema se ha realizado desde una IP francesa, lo cual es cierto, ya que actualmente estoy pasando las vacaciones de Navidad en casa de mis suegros. Ahora vamos a parar la máquina virtual, salvando su estado actual, para poder recuperarla posteriormente. La máquina actual no es una imagen persistente. Sin embargo está creada a partir de una imagen qcow2 manteniendo en un fichero aparte la información modificada sobre la imagen original (el "copy on write"). Vamos a verlo directamente sobre el datastore, y después realizaremos un "undeploy":

```
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h35
   221 customer users    centro65_vlan3_ runn    0    512M one-node1    0d 00h43
```

Como la imagen original usa un datastore especial, el fichero raíz está en el datastore 100, mientras que el fichero delta se encuentra en el datastore 0, dentro del directorio correspondiente al VM_id:

```
[oneadmin@one-node1 mytemplates]$ cd /var/lib/one/datastores
[oneadmin@one-node1 datastores]$ ll
total 32
drwxr-x---  4 oneadmin oneadmin  4096 Dec 23 01:41 0
drwxr-x---  2 oneadmin oneadmin  4096 Dec 23 01:11 1
drwxr-xr-x  2 oneadmin oneadmin  4096 Dec 23 01:11 100
drwxr-xr-x  2 oneadmin oneadmin  4096 Nov 15 01:23 2
drwx------. 2 oneadmin oneadmin 16384 May  9  2012 lost+found
[oneadmin@one-node1 datastores]$ cd 0/221/
[oneadmin@one-node1 221]$ ll
total 11900
-rw-rw-r-- 1 oneadmin oneadmin      866 Dec 23 01:41 deployment.0
-rw-r--r-- 1 oneadmin oneadmin 11927552 Dec 23 02:27 disk.0
-rw-r--r-- 1 oneadmin oneadmin   372736 Dec 23 01:41 disk.1
lrwxrwxrwx 1 oneadmin oneadmin       36 Dec 23 01:41 disk.1.iso ->
/var/lib/one/datastores/0/221/disk.1
[oneadmin@one-node1 221]$ qemu-img info disk.0
image: disk.0
file format: qcow2
virtual size: 40G (42949672960 bytes)
disk size: 11M
cluster_size: 65536
backing file: /var/lib/one/datastores/100/d660bcd3f0d64b37634e000e40d25238
```

Esto permite que el disco desplegado ocupe muy poco espacio. Ahora vamos a hacer el undeploy, y veremos que pasa a través de la consola VNC:

```
[oneadmin@one-node1 221]$ onevm undeploy 221
```

```
[oneadmin@one-node1 221]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h41
   221 customer users    centro65_vlan3_ unde    0    512M              0d 00h49
```

Vemos que no podemos acceder a la máquina, y que nos aparece en estado undeployed.

```
[oneadmin@one-node1 221]$ onevm show 221
VIRTUAL MACHINE 221 INFORMATION
ID                  : 221
NAME                : centro65_vlan3_1
USER                : customer1
GROUP               : users
STATE               : UNDEPLOYED
LCM_STATE           : LCM_INIT
RESCHED             : No
START TIME          : 12/23 01:41:19
END TIME            : -
DEPLOY ID           : one-221

VIRTUAL MACHINE MONITORING
USED CPU            : 0
NET_TX              : 27K
USED MEMORY         : 512M
NET_RX              : 29K

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    CentOS 6.5 non persistent           file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.120.2   02:00:c0:a8:78:02
                                           fe80::400:c0ff:fea8:7802

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START        TIME     PROLOG
  0 one-node1       undeploy         user   12/23 01:41:32   0d 00h48m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.120.2",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  NETWORK="YES",
  TARGET="hdb" ]
CPU="0.5"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="6121",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="4"
VMID="221"
```

Vamos a volver a reiniciar la máquina, con un resume:

```
[oneadmin@one-node1 221]$ onevm resume 221
[oneadmin@one-node1 221]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h46
   221 customer users    centro65_vlan3_ boot    0    512M one-node1    0d 00h55
[oneadmin@one-node1 221]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h46
   221 customer users    centro65_vlan3_ runn    0    512M one-node1    0d 00h55
```

Vemos que la máquina arranca de nuevo sin problemas. Vamos a "apagar" la máquina, desde dentro del sistema (es decir, sin usar el API de OpenNebula):

```
[jamontes@Tesla .ssh]$ ssh root@192.101.5.34 -p 2222
root@192.101.5.34's password:
Last login: Mon Dec 23 01:48:57 2013 from adijon-151-1-4-151.w83-194.abo.wanadoo.fr
[root@localhost ~]# init 0
[root@localhost ~]# Connection to 192.101.5.34 closed by remote host.
Connection to 192.101.5.34 closed.
[oneadmin@one-node1 221]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h51
   221 customer users    centro65_vlan3_ unkn    0    512M one-node1    0d 00h59
```

Como no hemos usado el API, la máquina aparece en estado unknown.

Vamos a probar a hacer un boot, para iniciar la máquina:

```
[oneadmin@one-node1 221]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 03h54
   221 customer users    centro65_vlan3_ runn    0    512M one-node1    0d 01h02
```

La máquina arranca perfectamente:

```
[jamontes@Tesla .ssh]$ ssh root@192.101.5.34 -p 2222
root@192.101.5.34's password:
Last login: Mon Dec 23 02:40:09 2013 from adijon-151-1-4-151.w83-194.abo.wanadoo.fr
[root@localhost ~]#
```

Con esto dejamos aclarado el tema de la persistencia.

4.3.5. Configurando una red privada para el usuario customer1

Vamos a dejar disponible la VLAN2 para el usuario customer1:

```
[oneadmin@one-node1 ~]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 oneadmin     oneadmin     VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          1
   5 customer1    users        VLAN3           -               R brvl3         2
   6 oneadmin     oneadmin     RTVLAN3         -               R brvl3         1
[oneadmin@one-node1 ~]$ onevnet show 1
VIRTUAL NETWORK 1 INFORMATION
ID             : 1
NAME           : VLAN2
USER           : oneadmin
GROUP          : oneadmin
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl2
VLAN           : Yes
PHYSICAL DEVICE: em4
VLAN ID        : 2
USED LEASES    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="10.132.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 10.132.120.1
```

```
IP_END          : 10.132.127.254

VIRTUAL MACHINES
```

Cambiamos los propietarios de la red:

```
[oneadmin@one-node1 ~]$ onevnet chown 1 2 1
[oneadmin@one-node1 ~]$ onevnet show 1
VIRTUAL NETWORK 1 INFORMATION
ID              : 1
NAME            : VLAN2
USER            : customer1
GROUP           : users
CLUSTER         : -
TYPE            : RANGED
BRIDGE          : brvl2
VLAN            : Yes
PHYSICAL DEVICE: em4
VLAN ID         : 2
USED LEASES     : 0

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="10.132.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START        : 10.132.120.1
IP_END          : 10.132.127.254

VIRTUAL MACHINES
```

Esta red será privada, sólo accesible por el usuario customer1, y no tendrá salida a internet.

4.3.6. Preparando el resto de plantillas para customer1

Ahora creamos nuevas plantillas para las instancias del tty-linuxV3 y del CentOS 6.5:

```
[oneadmin@one-node1 mytemplates]$ more ttyVL2_3_contextNP.tmpl
NAME="tty_linuxV3 VLANs 2 3 NP"
CPU="0.1"
DISK=[
  IMAGE_ID="12" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
```

```
MEMORY="64"
FEATURES=[
  ACPI="no" ]
NIC = [ NETWORK="VLAN3" ]
NIC = [ NETWORK="VLAN2" ]

CONTEXT=[
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
[oneadmin@one-node1 mytemplates]$ onetemplate create ttyVL2_3_contextNP.tmpl
ID: 5
[oneadmin@one-node1 mytemplates]$ onetemplate chown 5 2 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 5
TEMPLATE 5 INFORMATION
ID              : 5
NAME            : tty_linuxV3 VLANs 2 3 NP
USER            : customer1
GROUP           : users
REGISTER TIME   : 12/23 15:12:44

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.1"
DISK=[
  IMAGE_ID="12" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="VLAN3" ]
NIC=[
  NETWORK="VLAN2" ]
```

Vamos a hacer lo mismo con la plantilla del CentOS 6.5. Aprovechando que la imagen de CentOS soporta los drivers virtio para acceso a disco y red, vamos a configurar la plantilla:

```
[oneadmin@one-node1 mytemplates]$ more centos65VL2_3_contextNP.tmpl
NAME="Centos65_x86_64 VLANs 2 3 NP"
CPU="0.5"
DISK=[
  IMAGE_ID="5",
  DEV_PREFIX=vd,
  TARGET=vda ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
FEATURES=[
  ACPI="yes" ]
NIC = [
  NETWORK="VLAN3",
  MODEL=virtio ]
NIC = [
  NETWORK="VLAN2",
  MODEL=virtio ]

CONTEXT=[
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
[oneadmin@one-node1 mytemplates]$ onetemplate create centos65VL2_3_contextNP.tmpl
ID: 6
[oneadmin@one-node1 mytemplates]$ onetemplate chown 6 2 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 6
TEMPLATE 6 INFORMATION
ID              : 6
NAME            : Centos65_x86_64 VLANs 2 3 NP
USER            : customer1
GROUP           : users
REGISTER TIME   : 12/23 15:48:42

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.5"
DISK=[
  DEV_PREFIX="vd",
  IMAGE_ID="5",
  TARGET="vda" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
NIC=[
  MODEL="virtio",
  NETWORK="VLAN3" ]
NIC=[
  MODEL="virtio",
  NETWORK="VLAN2" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Vamos a instanciar una máquina de cada plantilla, para estar seguros que el usuario customer1 puede utilizarlas sin problema:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 5 --name "ttyV3 VL2_3 1" --user customer1
--password "1solomio"
VM ID: 222
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 6 --name "CentOS65 VL2_3 1" --user
customer1 --password "1solomio"
VM ID: 223
```

Vamos a comprobar que el XML con el dominio generado para KVM utiliza los drivers virtio:

```
[oneadmin@one-node1 mytemplates]$ cd /var/lib/one/datastores/0/223
[oneadmin@one-node1 223]$ ll
total 11132
-rw-rw-r-- 1 oneadmin oneadmin     1027 Dec 23 15:52 deployment.0
-rw-r--r-- 1 oneadmin oneadmin 11141120 Dec 23 16:00 disk.0
-rw-r--r-- 1 oneadmin oneadmin   372736 Dec 23 15:52 disk.1
lrwxrwxrwx 1 oneadmin oneadmin       36 Dec 23 15:52 disk.1.iso ->
/var/lib/one/datastores/0/223/disk.1
[oneadmin@one-node1 223]$ more deployment.0
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
        <name>one-223</name>
        <cputune>
                <shares>512</shares>
        </cputune>
        <memory>524288</memory>
        <os>
                <type arch='x86_64'>hvm</type>
                <boot dev='hd'/>
        </os>
        <devices>
                <emulator>/usr/libexec/qemu-kvm</emulator>
                <disk type='file' device='disk'>
                        <source file='/var/lib/one//datastores/0/223/disk.0'/>
                        <target dev='vda'/>
                        <driver name='qemu' type='qcow2' cache='none'/>
                </disk>
                <disk type='file' device='cdrom'>
                        <source file='/var/lib/one//datastores/0/223/disk.1'/>
                        <target dev='hda'/>
                        <readonly/>
                        <driver name='qemu' type='raw'/>
                </disk>
                <interface type='bridge'>
                        <source bridge='brvl3'/>
                        <mac address='02:00:c0:a8:78:04'/>
                        <model type='virtio'/>
                </interface>
                <interface type='bridge'>
                        <source bridge='brvl2'/>
                        <mac address='02:00:0a:84:78:d5'/>
                        <model type='virtio'/>
                </interface>
                <graphics type='vnc' listen='0.0.0.0' port='6123'/>
        </devices>
        <features>
                <acpi/>
        </features>
</domain>
```

Vemos que efectivamente está utilizado los drivers virtio para el acceso al disco y a los interfaces de red.

4.4. Modificando el scheduler para pruebas de rendimiento

Ahora vamos a modificar los parámetros del scheduler para poder invocar las máquinas virtuales a un ritmo más rápido del que tiene por defecto (una máquina por host => 2 máquinas cada 30 segundos). Para ello tenemos que modificar el fichero /etc/one/sched.conf

```
[root@one-node1 one]# diff sched.conf sched.conf.org
47c47
< SCHED_INTERVAL = 5
---
> SCHED_INTERVAL = 30
```

```
49,51c49,51
< MAX_VM       = 0
< MAX_DISPATCH = 300
< MAX_HOST     = 300
---
> MAX_VM       = 5000
> MAX_DISPATCH = 30
> MAX_HOST     = 1
53c53
< LIVE_RESCHEDS  = 1
---
> LIVE_RESCHEDS  = 0
```

Los datos modificados son realmente agresivos, pero es para hacer la prueba de rendimiento y ver como se comporta el sistema. Para que los cambios tengan efecto, tenemos que reiniciar el scheduler:

```
[oneadmin@one-node1 ~]$ ps -ef | grep mm_sched
oneadmin  9841 60046  0 16:46 pts/0    00:00:00 grep mm_sched
oneadmin 20083     1  0 Nov15 ?        02:44:14 /usr/bin/mm_sched
[oneadmin@one-node1 ~]$ kill -9 20083
[oneadmin@one-node1 ~]$ nohup /usr/bin/mm_sched >/dev/null 2>&1 &
[1] 10557
[oneadmin@one-node1 ~]$ ps -ef | grep mm_sched
oneadmin 10557 60046  0 16:48 pts/0    00:00:00 /usr/bin/mm_sched
oneadmin 10918 60046  0 16:49 pts/0    00:00:00 grep mm_sched
```

Vamos a comprobar los logs:

```
[root@one-node1 ~]# more /var/log/one/sched.log
Mon Dec 23 16:48:36 2013 [SCHED][I]: Init Scheduler Log system
Mon Dec 23 16:48:36 2013 [SCHED][I]: Starting Scheduler Daemon
----------------------------------------
     Scheduler Configuration File
----------------------------------------
DEFAULT_SCHED=POLICY=1
HYPERVISOR_MEM=0.1
LIVE_RESCHEDS=1
LOG=DEBUG_LEVEL=3,SYSTEM=file
MAX_DISPATCH=300
MAX_HOST=300
MAX_VM=0
ONED_PORT=2633
SCHED_INTERVAL=5
----------------------------------------
Mon Dec 23 16:48:36 2013 [SCHED][I]: Starting scheduler loop...
Mon Dec 23 16:48:36 2013 [SCHED][I]: Scheduler loop started.
```

4.4.1. Preparación de las pruebas de rendimiento

Bien, ya lo tenemos preparado. Ahora vamos a instanciar unas 1500 máquinas virtuales del tipo ttyV3. Como aún no tenemos asignadas las cuotas para el usuario customer1, podemos hacerlo sabiendo que tenemos disponible capacidad física suficiente entre los dos hosts:

```
[oneadmin@one-node1 ~]$ onetemplate instantiate 5 --name "ttyV3_%i" --user customer1 --password
"1solomio" -m 1500 -v
```

Vemos que las instancias son creadas inmediatamente una tras otra, hasta completar las 1500:

```
VMTEMPLATE 5: instantiated
VM ID: 1722
VMTEMPLATE 5: instantiated
VM ID: 1723
VMTEMPLATE 5: instantiated
[oneadmin@one-node1 ~]$
```

Por otra parte vemos que hay instancias que han fallado. Al ejecutar el comando “onevm show 846” nos aparece el siguiente error:

```
ERROR="Mon Dec 23 17:26:29 2013 : Error executing image transfer script: Error copying one-
node1:/var/lib/one/datastores/100/a39b7191c665bd2f57129a70b63838b3 to one-
node2:/var/lib/one//datastores/0/846/disk.0"
```

Es posible que haya que afinar los valores del scheduler para lanzar menos instancias cada vez. Esto también ocurre sobre el nodo1, al transferir la imagen ISO con el contenido de los parámetros de contexto:

```
ERROR="Mon Dec 23 17:26:45 2013 : Error executing image transfer script: Error copying context ISO to
one-node1:/var/lib/one//datastores/0/1204/disk.1"
```

La carga del sistema ha tenido un pico importante al invocar las máquinas de golpe. Esto es lo que vemos con el comando top en el host1:

```
top - 17:38:10 up 39 days, 14:47,  2 users,  load average: 0.91, 20.51, 20.84
Tasks: 1890 total,   4 running, 1885 sleeping,   0 stopped,   1 zombie
Cpu(s):  2.4%us,  3.7%sy,  0.0%ni, 93.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65951576k total, 47453992k used, 18497584k free,   435948k buffers
Swap: 67107832k total,        0k used, 67107832k free, 38137664k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
10557 oneadmin  20   0  262m 127m 3272 S 22.3  0.2   1:42.28 mm_sched
20082 oneadmin  20   0 1493m 112m 4500 S  7.5  0.2   2346:25 oned
12807 oneadmin  20   0  392m  34m 4468 S  4.6  0.1   0:48.01 qemu-kvm
20682 oneadmin  20   0  392m  34m 4468 S  4.6  0.1   0:49.36 qemu-kvm
 4745 oneadmin  20   0  392m  32m 4468 S  4.3  0.1   0:34.28 qemu-kvm
11252 oneadmin  20   0  392m  36m 4468 S  4.3  0.1   0:47.62 qemu-kvm
54351 oneadmin  20   0  392m  34m 4468 S  4.3  0.1   0:50.21 qemu-kvm
 8847 oneadmin  20   0  392m  34m 4468 S  3.9  0.1   0:47.45 qemu-kvm
14061 oneadmin  20   0  392m  34m 4468 S  3.9  0.1   0:46.89 qemu-kvm
26887 oneadmin  20   0  392m  34m 4468 S  3.9  0.1   0:51.18 qemu-kvm
```

Si lanzamos menos máquinas por vez y espaciamos un poco mas el ciclo (unos 10 segundos) no debería de haber problemas. Ahora que están todas las máquinas invocadas, podemos contar cuantas han fallado, cuantas están en proceso de arranque, y cuantas están ya en ejecución:

```
[oneadmin@one-node1 ~]$ onevm list | grep -c fail
482
[oneadmin@one-node1 ~]$ onevm list | grep -c boot
391
[oneadmin@one-node1 ~]$ onevm list | grep -c unk
205
[oneadmin@one-node1 ~]$ onevm list | grep -c runn
448
```

Vamos a esperar a que terminen de arrancar los que están en estado boot, y después veremos los que están en estado unknown. Al lanzar de nuevo un comando vemos que el host1 (donde está corriendo los procesos del front-end) está consumiendo demasiados recursos (como cuando lanzamos una bomba fork):

```
[oneadmin@one-node1 ~]$ onevm list | grep -c boot
-bash: fork: retry: Resource temporarily unavailable
-bash: fork: retry: Resource temporarily unavailable
-bash: fork: retry: Resource temporarily unavailable
^C
```

Y esto es lo que nos da el top:

```
top - 17:57:17 up 39 days, 15:06,  2 users,  load average: 328.21, 250.25, 143.51
Tasks: 2524 total,  63 running, 2460 sleeping,   0 stopped,   1 zombie
Cpu(s):  6.9%us, 49.5%sy,  3.3%ni, 28.5%id,  1.2%wa,  0.0%hi, 10.7%si,  0.0%st
Mem:  65951576k total, 55378964k used, 10572612k free,   466380k buffers
Swap: 67107832k total,        0k used, 67107832k free, 38521736k cached
```

Conclusiones:

- El sistema de polling de la versión 4.2 de OpenNebula no es eficiente ni escalable. La versión mas reciente (la 4.4, que ha salido hace unos días) ya soporta un sistema de pushing, que permite monitorizar hasta 25.000 máquinas virtuales cada 2 minutos.
- Aunque tengamos capacidad de computación suficiente en los hosts para albergar hasta unas 750 máquinas virtuales de 0.1 CPU cada una (tenemos 80 CPUs en cada host) a efectos prácticos, no debemos de tener en ejecución mas de 200 o 300 instancias por host.
- Debemos configurar los valores del scheduler que espacien mejor las operaciones. Los valores por defecto son bastante conservadores y pueden valer.

En el host2, podemos ver que anda mucho mejor de carga que su gemelo, al tener los recursos 100% dedicados a computación de las máquinas virtuales:

```
top - 18:16:46 up 40 days,  5:57,  1 user,  load average: 7.30, 25.28, 77.63
Tasks: 2574 total,   8 running, 2566 sleeping,   0 stopped,   0 zombie
Cpu(s):  8.2%us, 14.1%sy,  0.0%ni, 77.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65951576k total, 17525328k used, 48426248k free,   189668k buffers
Swap: 67107832k total,        0k used, 67107832k free,  1018216k cached
```

Podemos contar el número de instancias en ejecución a través del comando virsh:

```
[root@one-node2 ~]# virsh list | grep -c runn
433
```

Esto demuestra que el frontend debería de estar ejecutándose en otra máquina diferente a los hosts de computación. Al cabo de una hora aproximadamente de haber lanzado el experimento comprobamos el número de máquinas que se encuentra en ejecución en cada nodo:

```
[root@one-node1 ~]# virsh list --all | grep -c runn
474
[root@one-node2 ~]# virsh list --all | grep -c runn
433
```

Después de confirmar que los valores del scheduler actuales no permiten mantener el sistema estable, vamos a partir de cero de nuevo, borrando las instancias existentes:

```
[oneadmin@one-node1 ~]$ onevm delete 224..1723
```

El resultado es una hilera de lineas "Broken pipe". Tenemos que esperar a que el scheduler termine de limpiar las instancias, y el host1 quede más descargado. Ya está. Nos hemos quedado con las máquinas que teníamos antes de lanzar las 1500 instancias.

```
[oneadmin@one-node1 ~]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 22h42
   221 customer users    centro65_vlan3_ runn    0  533.4M one-node1    0d 19h50
   222 customer users    ttyV3 VL2_3 1   runn    5     64M one-node2    0d 05h40
   223 customer users    CentOS65 VL2_3  runn    0  659.3M one-node1    0d 05h40
```

Vamos a remplazar el fichero /etc/one/sched.conf por su original y a reiniciar el proceso mm_sched:

```
[root@one-node1 one]# cp sched.conf sched.conf.nogoodforscale
[root@one-node1 one]# cp sched.conf.org sched.conf
cp: overwrite `sched.conf'? y
```

Y lanzamos de nuevo el proceso desde el usuario oneadmin:

```
[oneadmin@one-node1 ~]$ ps -ef | grep mm_sched
oneadmin 10557 60046  9 16:48 pts/0    00:27:47 /usr/bin/mm_sched
oneadmin 39673 60046  0 21:52 pts/0    00:00:00 grep mm_sched
[oneadmin@one-node1 ~]$ kill -9 10557
[oneadmin@one-node1 ~]$ nohup /usr/bin/mm_sched >/dev/null 2>&1 &
[3] 39682
[2]   Killed                  nohup /usr/bin/mm_sched > /dev/null 2>&1
[oneadmin@one-node1 ~]$
[oneadmin@one-node1 ~]$ ps -ef | grep mm_sched
oneadmin 39682 60046  0 21:52 pts/0    00:00:00 /usr/bin/mm_sched
oneadmin 39781 60046  0 21:52 pts/0    00:00:00 grep mm_sched
```

Ya está. Ahora vamos a lanzar 240 máquinas virtuales de la plantilla 6 (centos 6.5 con dos interfaces de red, 0.5 CPU, y 512M de RAM). Esta vez no debería de haber problemas de despliegue al haber dejado los valores originales, y mantenernos por debajo del umbral de las 200 máquinas. No podemos lanzar mas de 120 a 125 máquinas por host porque sólo tenemos 64G de RAM en total.

```
[oneadmin@one-node1 ~]$ onetemplate instantiate 6 --name "CentOS65_%i" --user customer1 --password
"1solomio" -m 240 -v
VM ID: 1724
VMTEMPLATE 6: instantiated
VM ID: 1725
VMTEMPLATE 6: instantiated
VM ID: 1726
VMTEMPLATE 6: instantiated
VM ID: 1727
VMTEMPLATE 6: instantiated
VM ID: 1728
VMTEMPLATE 6: instantiated
VM ID: 1729
........
VMTEMPLATE 6: instantiated
VM ID: 1962
VMTEMPLATE 6: instantiated
VM ID: 1963
VMTEMPLATE 6: instantiated
[oneadmin@one-node1 ~]$
```

Ahora veremos cómo las máquinas se van instanciando, a un ritmo de 4 por minuto (los ciclos son de 30s, y en cada uno de ellos se invoca una instancia por host. Los cálculos nos dicen que al cabo de una hora, las 240 instancias quedarán en ejecución. Podemos ver cómo van avanzando con el comando onevm top, aunque como se trata de 240 máquinas, se nos va de la pantalla. Sin embargo si que podemos ver como se van arrancando las máquinas con un simple list:

```
[oneadmin@one-node1 ~]$ onevm list | more
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
   215 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 23h15
   221 customer users    centro65_vlan3_ runn    0    512M one-node1    0d 20h23
   222 customer users    ttyV3 VL2_3 1   runn    4     64M one-node2    0d 06h13
   223 customer users    CentOS65 VL2_3  runn    0    512M one-node1    0d 06h13
  1724 customer users    CentOS65_0      runn    0    512M one-node2    0d 00h03
  1725 customer users    CentOS65_1      runn    0    512M one-node1    0d 00h03
  1726 customer users    CentOS65_2      runn    0    512M one-node2    0d 00h03
  1727 customer users    CentOS65_3      runn    0    512M one-node1    0d 00h03
  1728 customer users    CentOS65_4      runn    0    512M one-node2    0d 00h03
  1729 customer users    CentOS65_5      runn    0    512M one-node1    0d 00h03
  1730 customer users    CentOS65_6      runn    0    512M one-node2    0d 00h03
  1731 customer users    CentOS65_7      runn    2    512M one-node1    0d 00h03
  1732 customer users    CentOS65_8      runn    0    512M one-node2    0d 00h03
  1733 customer users    CentOS65_9      runn    0    512M one-node1    0d 00h03
  1734 customer users    CentOS65_10     runn   11    512M one-node2    0d 00h03
  1735 customer users    CentOS65_11     runn   10    512M one-node1    0d 00h03
  1736 customer users    CentOS65_12     runn  100    512M one-node2    0d 00h03
  1737 customer users    CentOS65_13     runn   99    512M one-node1    0d 00h03
  1738 customer users    CentOS65_14     runn    0      0K one-node2    0d 00h03
  1739 customer users    CentOS65_15     runn    0      0K one-node1    0d 00h03
```

Para confirmar el tiempo de despliegue de cada máquina, podemos verlo sobre el valor del campo "prolog"

```
[oneadmin@one-node1 ~]$ onevm show 1724
VIRTUAL MACHINE 1724 INFORMATION
ID                  : 1724
NAME                : CentOS65_0
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node2
START TIME          : 12/23 22:01:01
END TIME            : -
DEPLOY ID           : one-1724

VIRTUAL MACHINE MONITORING
NET_TX              : 2K
NET_RX              : 32K
USED CPU            : 0
USED MEMORY         : 512M

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 vda    CentOS 6.5 non persistent           file   NO       -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.125.224 02:00:c0:a8:7d:e0
                                           fe80::400:c0ff:fea8:7de0
  1 VLAN2                 yes brvl2        10.132.126.177  02:00:0a:84:7e:b1
                                           fe80::400:aff:fe84:7eb1

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node2       none             none  12/23 22:01:09   0d 00h05m   0h00m00s

VIRTUAL MACHINE TEMPLATE
```

```
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.125.224",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  ETH1_IP="10.132.126.177",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="10.132.120.0/21",
  NETWORK="YES",
  TARGET="hda" ]
CPU="0.5"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="7624",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="6"
VMID="1724"
```

Vemos que el tiempo de preparar la nueva imagen (PROLOG) con el driver qcow es prácticamente instantáneo, incluso estando sobre el host2. Vemos otro ejemplo en el host1:

```
[oneadmin@one-node1 ~]$ onevm show 1725
VIRTUAL MACHINE 1725 INFORMATION
ID                    : 1725
NAME                  : CentOS65_1
USER                  : customer1
GROUP                 : users
STATE                 : ACTIVE
LCM_STATE             : RUNNING
RESCHED               : No
HOST                  : one-node1
START TIME            : 12/23 22:01:01
END TIME              : -
DEPLOY ID             : one-1725

VIRTUAL MACHINE MONITORING
NET_RX                : 44K
USED MEMORY           : 512M
NET_TX                : 2K
```

```
USED CPU            : 0

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 vda    CentOS 6.5 non persistent           file   NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.125.225 02:00:c0:a8:7d:e1
                                           fe80::400:c0ff:fea8:7de1
  1 VLAN2                 yes brvl2        10.132.126.178  02:00:0a:84:7e:b2
                                           fe80::400:aff:fe84:7eb2

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START        TIME     PROLOG
  0 one-node1       none             none  12/23 22:01:09   0d 00h08m   0h00m01s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.125.225",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  ETH1_IP="10.132.126.178",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="10.132.120.0/21",
  NETWORK="YES",
  TARGET="hda" ]
CPU="0.5"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="7625",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="6"
VMID="1725"
```

Se puede comprobar que el tiempo de despliegue de la imagen de 40G con el driver qcow2 es de alrededor de un segundo. Otras de las bondades del driver qcow2 es que permite realizar snapshots y regresiones, incluso con la máquina en activo.

Después de una hora, podremos comprobar que las máquinas virtuales se han instanciado correctamente, y que no queda ninguna máquina en estado fail o unknown. También podremos comprobar que las instancias se han repartido de forma equilibrada entre los dos hosts, que era el objetivo perseguido con la configuración del scheduler.

Sin embargo nos hemos quedado cortos en nuestros cálculos con respecto a la memoria utilizada, y nos quedan 17 máquinas en estado pendiente debido a que no quedan recursos de memoria disponibles entre los dos hosts.

La explicación de esto, es que en el fichero /etc/one/sched.conf tenemos lo siguiente:

```
HYPERVISOR_MEM = 0.1
```

Lo cual significa que podemos asignar hasta un 90% de la memoria disponible en cada host. Si echamos un vistazo a la ocupación de los dos host, veremos que estamos justo en el límite de la memoria:

```
[oneadmin@one-node1 ~]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -          113  5650 / 8000 (70%) 56.5G / 62.9G (89% update
   1 one-node2       -          114  5630 / 8000 (70%) 56.2G / 62.9G (89% update
```

Y si miramos alguna de las instancias que están pendientes de instanciarse, podremos ver la causa de por qué se han quedado pendientes:

```
[oneadmin@one-node1 ~]$ onevm show 1951
VIRTUAL MACHINE 1951 INFORMATION
ID                  : 1951
NAME                : CentOS65_227
USER                : customer1
GROUP               : users
STATE               : PENDING
LCM_STATE           : LCM_INIT
RESCHED             : No
START TIME          : 12/23 22:01:06
END TIME            : -
DEPLOY ID           : -

VIRTUAL MACHINE MONITORING
USED CPU            : 0
NET_TX              : 0K
USED MEMORY         : 0K
NET_RX              : 0K

PERMISSIONS
OWNER               : um-
```

```
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 vda    CentOS 6.5 non persistent           file   NO       -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 VLAN3                 yes brvl3        192.168.126.195 02:00:c0:a8:7e:c3
                                           fe80::400:c0ff:fea8:7ec3
  1 VLAN2                 yes brvl2        10.132.127.148  02:00:0a:84:7f:94
                                           fe80::400:aff:fe84:7f94

USER TEMPLATE
SCHED_MESSAGE="Mon Dec 23 23:13:09 2013 : No host with enough capacity to deploy the VM"

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.168.120.1",
  ETH0_IP="192.168.126.195",
  ETH0_MASK="255.255.248.0",
  ETH0_NETWORK="192.168.120.0/21",
  ETH1_IP="10.132.127.148",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="10.132.120.0/21",
  NETWORK="YES",
  TARGET="hda" ]
CPU="0.5"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="7851",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="6"
VMID="1951"
```

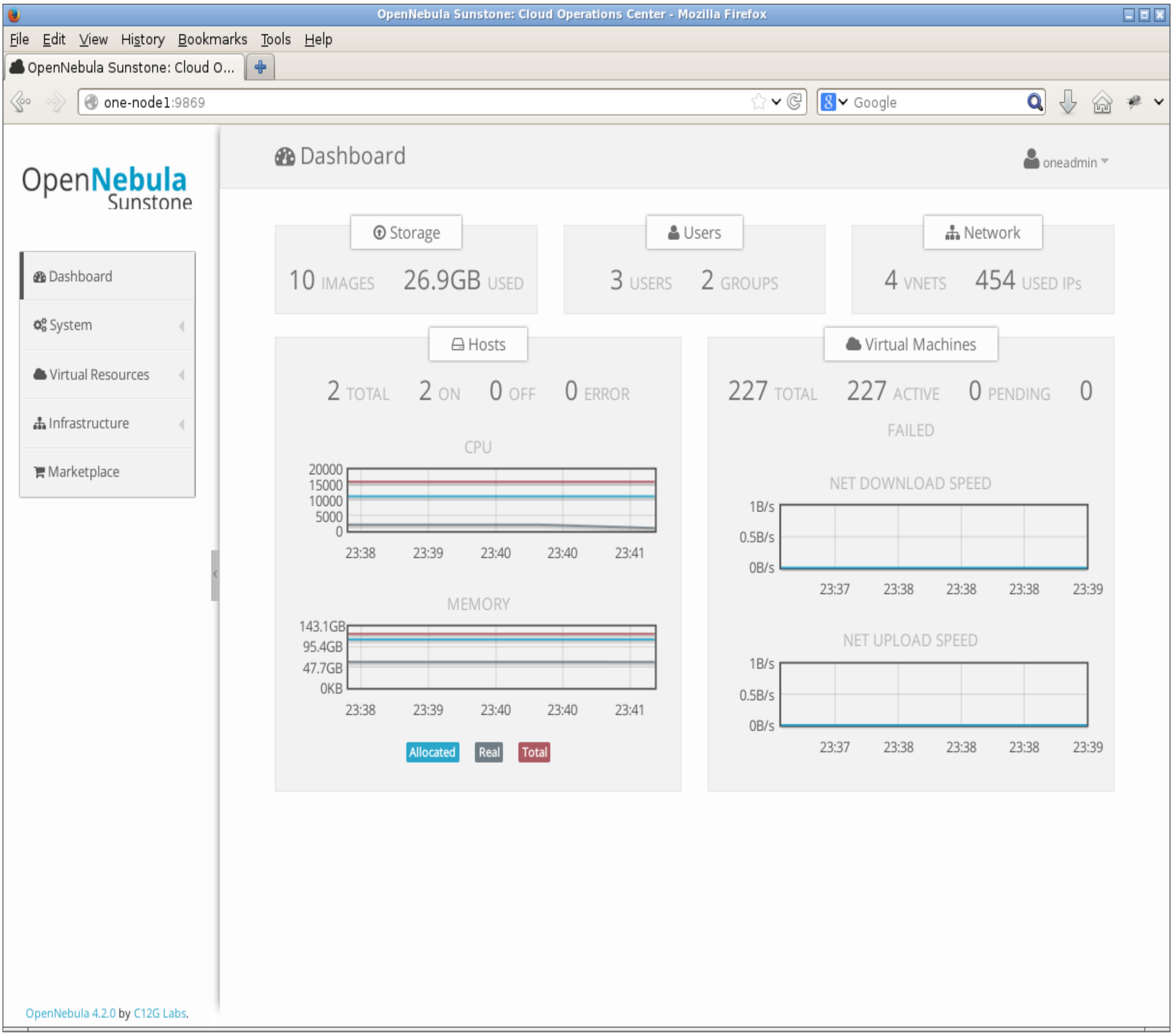
Esa es la explicación. Con estos cálculos, tenemos 6.4G de RAM reservados en cada host. Como es una medida prudente vamos a respetarla, y a eliminar las instancias pendientes:

```
[oneadmin@one-node1 ~]$ onevm delete 1947..1963
```

Con eso lo dejamos en 227 instancias en ejecución:

```
[oneadmin@one-node1 ~]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -          113  5650 / 8000 (70%) 56.5G / 62.9G (89% update
   1 one-node2       -          114  5630 / 8000 (70%) 56.2G / 62.9G (89% update
```

Hemos capturado una imagen de cómo aparecen las 227 imágenes recién instanciadas en el Dashboard del Sunstone:



5. Instalación de Ganglia como sistema de monitorización

Ahora vamos a instalar Ganglia, para poder monitorizar la plataforma. Para balancear la carga entre los dos hosts, vamos a instalar el recolector y el frontend web en el host2. Como tenemos el repositorio EPEL en ambos hosts, la instalación es directa y sencilla. Comenzamos con el host1:

```
[root@one-node1 ~]# yum -y install ganglia ganglia-gmond
```

Y en el host2 instalamos el monitor, el recolector, y el fontend:

```
[root@one-node2 ~]# yum -y install ganglia ganglia-gmond ganglia-gmetad ganglia-web
```

Ahora comenzamos con la configuración. En el host1 configuramos lo siguiente:

```
[root@one-node1 ~]# cd /etc/ganglia/
[root@one-node1 ganglia]# ll
total 12
drwxr-xr-x 2 root root 4096 Dec 26 10:54 conf.d
-rw-r--r-- 1 root root 7788 Jul 15  2012 gmond.conf
```

Estos son los cambios realizados sobre el fichero original:

```
[root@one-node1 ganglia]# diff gmond.conf gmond.conf.org
23c23
<   name = "OpenNebula"
---
>   name = "unspecified"
31c31
<   location = "one-node1"
---
>   location = "unspecified"
37c37
<   bind_hostname = yes  # Highly recommended, soon to be default.
---
>   #bind_hostname = yes # Highly recommended, soon to be default.
43c43
<   host = one-node2
---
>   mcast_join = 239.2.11.71
47a48,60
> /* You can specify as many udp_recv_channels as you like as well. */
> udp_recv_channel {
>   mcast_join = 239.2.11.71
>   port = 8649
>   bind = 239.2.11.71
> }
>
> /* You can specify as many tcp_accept_channels as you like to share
>    an xml description of the state of the cluster */
> tcp_accept_channel {
>   port = 8649
> }
>
```

O mejor mostrado:

```
cluster {
  name = "OpenNebula"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}

host {
  location = "one-node1"
}

udp_send_channel {
  bind_hostname = yes  # Highly recommended, soon to be default.
                       # This option tells gmond to use a source address
                       # that resolves to the machine's hostname.  Without
                       # this, the metrics may appear to come from any
                       # interface and the DNS names associated with
                       # those IPs will be used to create the RRDs.
  host = one-node2
  port = 8649
  ttl = 1
}
```

Con esos cambios le decimos que el nombre del cluster será "OpenNebula", que debe de mandar los paquetes en modo unicast al nodo one-node2, identificándose como one-node1, y que no use el modo multicast ni recoja información proveniente de otros nodos. Ahora reiniciamos el servicio y lo dejamos preparado para que arranque automáticamente:

```
[root@one-node1 ganglia]# service gmond start
Starting GANGLIA gmond:                                    [  OK  ]
[root@one-node1 ganglia]# service gmond status
gmond (pid 51834) is running...
[root@one-node1 ganglia]# chkconfig gmond on
[root@one-node1 ganglia]# chkconfig --list  gmond
gmond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Con el host2 hacemos lo mismo:

```
[root@one-node2 ~]# cd /etc/ganglia/
[root@one-node2 ganglia]# ll
total 32
drwxr-xr-x 2 root root   4096 Dec 26 10:57 conf.d
-rw-r--r-- 1 root root   4660 Jul 15  2012 conf.php
-rw-r--r-- 1 root root   5238 Jul 15  2012 gmetad.conf
-rw-r--r-- 1 root root   7788 Jul 15  2012 gmond.conf
-rw-r----- 1 root apache 1222 Feb 17  2010 private_clusters
```

```
[root@one-node2 ganglia]# cp gmond.conf gmond.conf.org
[root@one-node2 ganglia]# diff gmond.conf gmond.conf.org
23c23
<   name = "OpenNebula"
---
>   name = "unspecified"
31c31
<   location = "one-node2"
---
>   location = "unspecified"
37c37
<   bind_hostname = yes # Highly recommended, soon to be default.
---
>   #bind_hostname = yes # Highly recommended, soon to be default.
43c43
<   host = one-node2
---
>   mcast_join = 239.2.11.71
49a50
>   mcast_join = 239.2.11.71
50a52
>   bind = 239.2.11.71
```

O mejor mostrado:

```
cluster {
  name = "OpenNebula"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}

host {
  location = "one-node2"
}

udp_send_channel {
  bind_hostname = yes # Highly recommended, soon to be default.
                       # This option tells gmond to use a source address
                       # that resolves to the machine's hostname.  Without
                       # this, the metrics may appear to come from any
                       # interface and the DNS names associated with
                       # those IPs will be used to create the RRDs.
  host = one-node2
  port = 8649
  ttl = 1
}
```

```
udp_recv_channel {
  port = 8649
}

tcp_accept_channel {
  port = 8649
}
```

Con eso le indicamos también que reciba la información del host1. Gracias a que tenemos todo bloqueado por iptables, excepto la LAN de servicio, no corremos riesgos de que nuestra información del cluster quede expuesta desde el exterior. Ahora configuramos el daemon recolector de estadísticas:

```
[root@one-node2 ganglia]# cp gmetad.conf gmetad.conf.org
[root@one-node2 ganglia]# diff gmetad.conf gmetad.conf.org
39,41c39
< #data_source "my cluster" localhost
<
< data_source "OpenNebula" 60 localhost
---
> data_source "my cluster" localhost
```

O mejor mostrado: la única línea que tenemos que configurar es esta:

```
data_source "OpenNebula" 60 localhost
```

Con eso le decimos que obtenga por polling las estadísticas de los dos hosts, recolectada por el host2, cada 60 segundos. Ahora nos queda la parte web:

En el fichero /etc/hosts añadimos un alias para el servidor web. Por motivos de seguridad nos conectaremos siempre desde localhost, a través de un tunel ssh.

```
127.0.0.1   localhost one-cluster
```

Y en el fichero /etc/httpd/conf/http.conf definiremos el servername con ese alias:

```
Listen one-cluster:80
ServerName one-cluster:80
```

Como vamos a conectarnos desde la ip de loopback, no necesitamos cambiar el fichero /etc/httpd/conf.d/ganglia.conf . Ahora sólo nos queda reiniciar los daemons en el host2:

```
[root@one-node2 ganglia]# service gmond status
gmond (pid 13839) is running...
[root@one-node2 ganglia]# service gmetad start
Starting GANGLIA gmetad:                                   [  OK  ]
[root@one-node2 ganglia]# service gmetad status
gmetad (pid 13887) is running...
[root@one-node2 ganglia]# service httpd start
Starting httpd:                                            [  OK  ]
[root@one-node2 ganglia]# service httpd status
httpd (pid  14650) is running...
```

Y con eso ya lo tenemos todo montado. Vamos a comprobar que en el host2 se van recolectando los datos de ambos hosts:

```
[root@one-node2 ~]# cd /var/lib/ganglia/rrds/OpenNebula
[root@one-node2 OpenNebula]# ll
total 12
drwxr-xr-x 2 ganglia root 4096 Dec 26 14:40 one-node1
drwxr-xr-x 2 ganglia root 4096 Dec 26 14:40 one-node2
drwxr-xr-x 2 ganglia root 4096 Dec 26 14:40 __SummaryInfo__
```

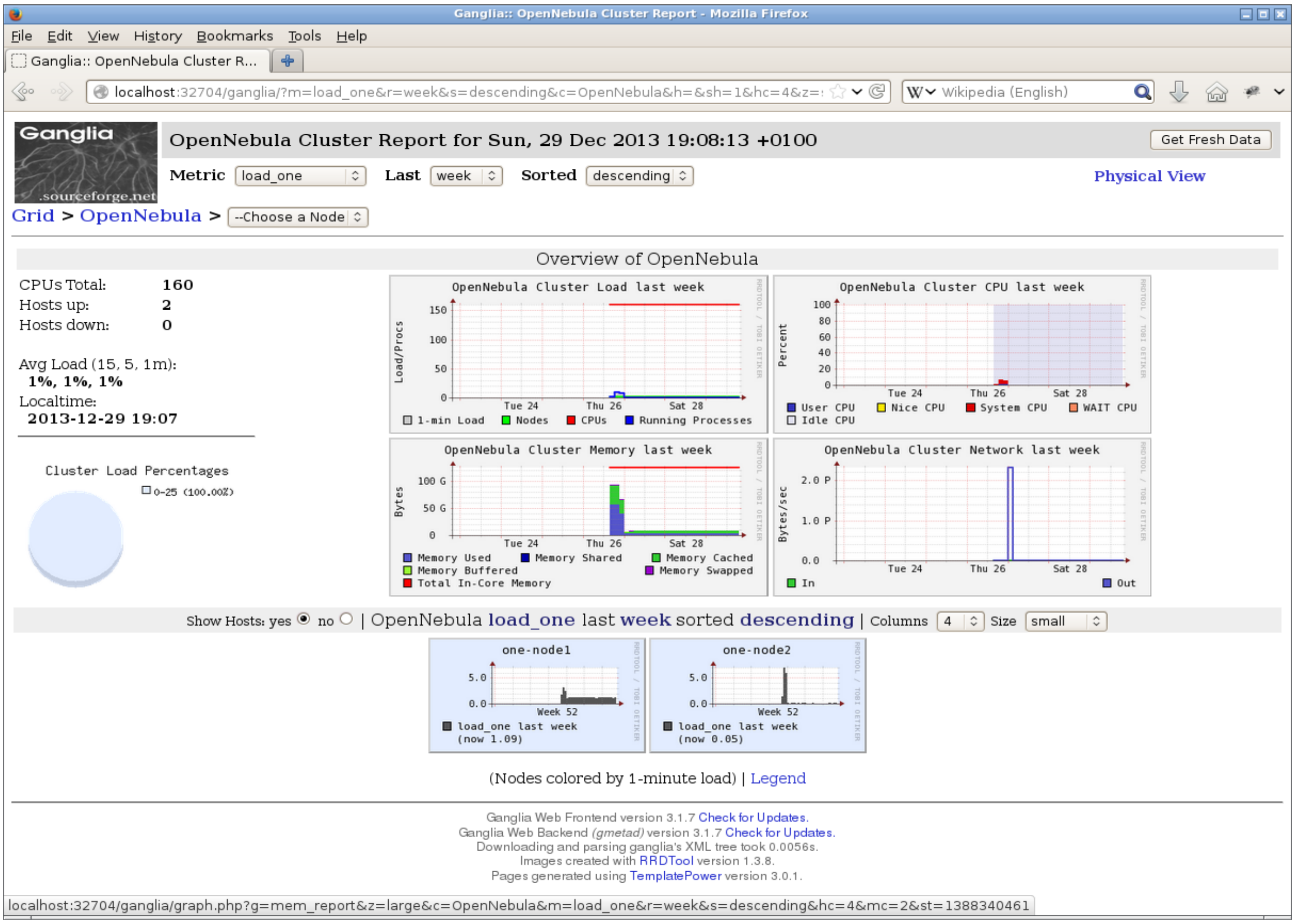
Bien, ahora vamos a conectarnos a la página web a través del túnel ssh, en el host2.

```
[jamontes@Tesla ~]$ ssh -L 32704:localhost:80 root@one-node2 -p 2222
bind: Cannot assign requested address
Last login: Thu Dec 26 10:33:50 2013 from 83.194.241.151
[root@one-node2 ~]#
```

Y ahora sólo nos queda conectarnos desde nuestro PC a la url:

```
http:localhost:32704/ganglia
```

Y obtenemos las gráficas de nuestro cluster. Aquí se puede apreciar una captura de imagen de la pantalla principal del cluster de Ganglia, con los dos nodos que lo componen:



Ganglia:: OpenNebula Cluster Report - Mozilla Firefox

localhost:32704/ganglia/?m=load_one&r=week&s=descending&c=OpenNebula&h=&sh=1&hc=4&z=:

OpenNebula Cluster Report for Sun, 29 Dec 2013 19:08:13 +0100

Metric load_one Last week Sorted descending

Grid > OpenNebula > --Choose a Node

Overview of OpenNebula

CPUs Total: 160
Hosts up: 2
Hosts down: 0

Avg Load (15, 5, 1m):
1%, 1%, 1%
Localtime:
2013-12-29 19:07

Show Hosts: yes ● no ○ | OpenNebula load_one last week sorted descending | Columns 4 Size small

(Nodes colored by 1-minute load) | Legend

Ganglia Web Frontend version 3.1.7 Check for Updates.
Ganglia Web Backend *(gmetad)* version 3.1.7 Check for Updates.
Downloading and parsing ganglia's XML tree took 0.0056s.
Images created with RRDTool version 1.3.8.
Pages generated using TemplatePower version 3.0.1.

localhost:32704/ganglia/graph.php?g=mem_report&z=large&c=OpenNebula&m=load_one&r=week&s=descending&hc=4&mc=2&st=1388340461

Todo ha quedado correctamente configurado. Con esto dejamos por finalizado el tema de la configuración de la monitorización por Ganglia.

6. Creación de un usuario adicional

Ahora vamos a crear otro usuario, definir los permisos y las plantillas, etc...

Tenemos que demostrar que los recursos asignados a cada usuario quedan completamente aislados entre si, gracias a los permisos que tiene cada uno sobre las plantillas, redes, etc...

Crearemos un usuario customer2, que pertenezca al mismo grupo que customer1:

```
[oneadmin@one-node1 ~]$ oneuser create customer2 2solomio
ID: 3
[oneadmin@one-node1 ~]$ oneuser list
  ID NAME            GROUP      AUTH           VMS            MEMORY         CPU
   0 oneadmin        oneadmin   core             -                 -           -
   1 serveradmin     oneadmin   server_c         -                 -           -
   2 customer1       users      core     226 /   0  112.6G /      0M 112.6 / 0.0
   3 customer2       users      core             -                 -           -
[oneadmin@one-node1 ~]$ oneuser show 3
USER 3 INFORMATION
ID              : 3
NAME            : customer2
GROUP           : users
PASSWORD        : be8d85c4d10c6159909efa8f1bf6f12081c743a2
AUTH_DRIVER     : core
ENABLED         : Yes

USER TEMPLATE
TOKEN_PASSWORD="3db37849d1644957dd8fdf3030b9fee2501fa991"

RESOURCE USAGE & QUOTAS
```

Ahora crearemos unas cuantas plantillas para que pueda usar un par de redes, tenga un router de salida a internet, etc...

Para este usuario usaremos la VLAN 5 como red interna con salida a internet, y la VLAN 6 como red privada de interconexión entre las máquinas del cliente. Aunque OpenNebula genera las MACs de los equipos en función de las IPs asignadas, al estar las LANs completamente aisladas mediante encapsulación 802.1Q, podemos repetir las IPs/MACs sin problemas de colisión de MACs, MAC/IP spoofing, etc. entre las redes creadas y utilizadas por cada usuario.

Vamos a definir las plantillas de las redes, modificando el número de VLAN para aislar los usuarios entre si. Definimos la VLAN privada para este cliente (VLAN 6) a partir de la VLAN2 creada para el anterior:

```
[oneadmin@one-node1 mytemplates]$ more priv_vlan2.net
NAME = "VLAN2"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 2
BRIDGE  = "brvl2"
```

```
NETWORK_ADDRESS = "10.132.120.0/21"
[oneadmin@one-node1 mytemplates]$ cp priv_vlan2.net priv_vlan6.net
[oneadmin@one-node1 mytemplates]$ vi priv_vlan6.net
[oneadmin@one-node1 mytemplates]$ more priv_vlan6.net
NAME = "VLAN6"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 6
BRIDGE  = "brvl6"

NETWORK_ADDRESS = "10.132.120.0/21"
[oneadmin@one-node1 mytemplates]$ onevnet create priv_vlan6.net
ID: 7
[oneadmin@one-node1 mytemplates]$ onevnet chown 7 3 1
[oneadmin@one-node1 mytemplates]$ onevnet show 7
VIRTUAL NETWORK 7 INFORMATION
ID              : 7
NAME            : VLAN6
USER            : customer2
GROUP           : users
CLUSTER         : -
TYPE            : RANGED
BRIDGE          : brvl6
VLAN            : Yes
PHYSICAL DEVICE: em4
VLAN ID         : 6
USED LEASES     : 0

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="10.132.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START        : 10.132.120.1
IP_END          : 10.132.127.254

VIRTUAL MACHINES
```

Así de sencillo. Ahora vamos a crear de igual manera la VLAN interna con salida a internet, a través de una instancia router dedicada para este cliente:

```
[oneadmin@one-node1 mytemplates]$ more priv_vlan3.net
NAME = "VLAN3"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brvl3"

NETWORK_ADDRESS = "192.168.120.0/21"
GATEWAY = "192.168.120.1"
DNS = "8.8.8.8 8.8.4.4"
IP_START = "192.168.120.1"
IP_END = "192.168.127.254"
[oneadmin@one-node1 mytemplates]$ cp priv_vlan3.net priv_vlan5.net
[oneadmin@one-node1 mytemplates]$ vi priv_vlan5.net
[oneadmin@one-node1 mytemplates]$ more priv_vlan5.net
NAME = "VLAN5"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 5
BRIDGE  = "brvl5"

NETWORK_ADDRESS = "192.168.120.0/21"
GATEWAY = "192.168.120.1"
DNS = "8.8.8.8 8.8.4.4"
IP_START = "192.168.120.1"
IP_END = "192.168.127.254"
[oneadmin@one-node1 mytemplates]$ onevnet create priv_vlan5.net
ID: 8
[oneadmin@one-node1 mytemplates]$ onevnet chown 8 3 1
```

Bloqueamos la IP usada como default gateway para que no pueda asignarla a ninguna VM:

```
[oneadmin@one-node1 mytemplates]$ onevnet hold 8 192.168.120.1
[oneadmin@one-node1 mytemplates]$ onevnet show 8
VIRTUAL NETWORK 8 INFORMATION
ID             : 8
NAME           : VLAN5
USER           : customer2
GROUP          : users
CLUSTER        : -
TYPE           : RANGED
BRIDGE         : brvl5
VLAN           : Yes
```

```
PHYSICAL DEVICE: em4
VLAN ID        : 5
USED LEASES    : 1

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

VIRTUAL NETWORK TEMPLATE
DNS="8.8.8.8 8.8.4.4"
GATEWAY="192.168.120.1"
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START       : 192.168.120.1
IP_END         : 192.168.127.254

LEASES ON HOLD
LEASE=[ MAC="02:00:c0:a8:78:01", IP="192.168.120.1", IP6_LINK="fe80::400:c0ff:fea8:7801", USED="1",
VID="-1" ]

VIRTUAL MACHINES

    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
```

Ahora creamos las plantillas para la instancia del router de salida a internet:

```
[oneadmin@one-node1 mytemplates]$ more router_vlan3.net
NAME = "RTVLAN3"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 3
BRIDGE  = "brvl3"

NETWORK_ADDRESS = "192.168.120.0/21"
[oneadmin@one-node1 mytemplates]$ vi router_vlan5.net
[oneadmin@one-node1 mytemplates]$ more router_vlan5.net
NAME = "RTVLAN5"
TYPE = "RANGED"
PHYDEV = "em4"
VLAN = "YES"
VLAN_ID = 5
BRIDGE  = "brvl5"
```

```
NETWORK_ADDRESS = "192.168.120.0/21"
[oneadmin@one-node1 mytemplates]$ onevnet create router_vlan5.net
ID: 9
[oneadmin@one-node1 mytemplates]$ onevnet show 9
VIRTUAL NETWORK 9 INFORMATION
ID              : 9
NAME            : RTVLAN5
USER            : oneadmin
GROUP           : oneadmin
CLUSTER         : -
TYPE            : RANGED
BRIDGE          : brvl5
VLAN            : Yes
PHYSICAL DEVICE: em4
VLAN ID         : 5
USED LEASES     : 0

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

VIRTUAL NETWORK TEMPLATE
NETWORK_ADDRESS="192.168.120.0/21"
NETWORK_MASK="255.255.248.0"

RANGE
IP_START        : 192.168.120.1
IP_END          : 192.168.127.254

VIRTUAL MACHINES
```

Y por último creamos la plantilla para la instancia del router, a partir de la plantilla empleada para el otro router. Podemos clonar y actualizar la plantilla, o bien modificar el fichero de la plantilla y crearla. Es cuestión de gustos, ya que el resultado final es el mismo.

```
[oneadmin@one-node1 mytemplates]$ cp router_vlan3.tmpl router_vlan5.tmpl
[oneadmin@one-node1 mytemplates]$ vi router_vlan5.tmpl
[oneadmin@one-node1 mytemplates]$ more router_vlan5.tmpl
NAME="RouterVLAN5"
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="128"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
FEATURES=[
  ACPI="yes" ]

NIC = [ NETWORK="Internet" ]
NIC = [
  NETWORK="RTVLAN5",
  IP="192.168.120.1" ]

CONTEXT=[
  TARGET        = "hdb",
  NETWORK       = "YES",
  SSH_PUBLIC_KEY = "ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  PRIVNET       = "$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET        = "$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  TEMPLATE      = "$TEMPLATE",
  DHCP          = "NO",
  RADVD         = "NO",
  FORWARDING    = "2222:192.168.120.2:22" ]
```

Nota: recordemos que la instancia router solo podrá ser manipulada por el administrador del sistema, porque así lo hemos establecido como política de administración del cluster.

```
[oneadmin@one-node1 mytemplates]$ vi router_vlan5.tmpl
[oneadmin@one-node1 mytemplates]$ onetemplate create router_vlan5.tmpl
ID: 7
[oneadmin@one-node1 mytemplates]$ onetemplate show 7
TEMPLATE 7 INFORMATION
ID             : 7
NAME           : RouterVLAN5
USER           : oneadmin
GROUP          : oneadmin
REGISTER TIME  : 12/26 17:49:54

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---
```

```
TEMPLATE CONTENTS
CONTEXT=[
  DHCP="NO",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",
  PRIVNET="$NETWORK[TEMPLATE, NETWORK=\"RTVLAN3\"]",
  PUBNET="$NETWORK[TEMPLATE, NETWORK=\"Internet\"]",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",
  TEMPLATE="$TEMPLATE" ]
CPU="0.2"
DISK=[
  IMAGE_ID="8" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="128"
NIC=[
  NETWORK="Internet" ]
NIC=[
  IP="192.168.120.1",
  NETWORK="RTVLAN5" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Instanciamos el router:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 7 --name "RouterVLAN5"
VM ID: 1964
[oneadmin@one-node1 mytemplates]$ onevm list
.........
  1940 customer users    CentOS65_216    runn    0    512M one-node2    2d 19h52
  1941 customer users    CentOS65_217    runn    0    512M one-node1    2d 19h52
  1942 customer users    CentOS65_218    runn    0    512M one-node2    2d 19h52
  1943 customer users    CentOS65_219    runn    1    512M one-node1    2d 19h52
  1944 customer users    CentOS65_220    runn    0    512M one-node2    2d 19h52
  1945 customer users    CentOS65_221    runn    0    512M one-node1    2d 19h52
  1946 customer users    CentOS65_222    runn    0    512M one-node2    2d 19h52
  1964 oneadmin oneadmin RouterVLAN5     boot    0      0K one-node2    0d 00h01
```

```
[oneadmin@one-node1 mytemplates]$ onevm show 1964
VIRTUAL MACHINE 1964 INFORMATION
ID                  : 1964
NAME                : RouterVLAN5
USER                : oneadmin
GROUP               : oneadmin
STATE               : ACTIVE
LCM_STATE           : BOOT
RESCHED             : No
HOST                : one-node2
START TIME          : 12/26 17:51:54
END TIME            : -
DEPLOY ID           : -

VIRTUAL MACHINE MONITORING
USED CPU            : 0
NET_RX              : 0K
USED MEMORY         : 0K
NET_TX              : 0K

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                               TYPE SAVE SAVE_AS
  0 hda    virtual_router qcow2 non persistent file   NO       -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet               no brp0         192.101.5.35    02:00:d4:e7:05:23
                                           fe80::400:d4ff:fee7:523
  1 RTVLAN5               yes brvl5        192.168.120.1   02:00:c0:a8:78:01
                                           fe80::400:c0ff:fea8:7801

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node2       none             none  12/26 17:52:09   0d 00h02m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DHCP="NO",
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.101.5.62",
  ETH0_IP="192.101.5.35",
  ETH0_MASK="255.255.255.224",
  ETH0_NETWORK="192.101.5.32/27",
  ETH1_IP="192.168.120.1",
  ETH1_MASK="255.255.248.0",
  ETH1_NETWORK="192.168.120.0/21",
  FORWARDING="2222:192.168.120.2:22",
  NETWORK="YES",

PUBNET="PFZORVQ+PElEPjI8L0lEPjxVSUQ+MDwvVUlEPjxHSUQ+MDwvR0lEPjxVTkFNRT5vbmVhZG1pbjwvVU5BTUU+PEdOQU1FP
m9uZWFkbWluPC9HTkFNRT48TkFNRT5JbnRlcm5ldDwvTkFNRT48UEVSTUlTU0lPTlM+PE9XTkVSX1U+MTwvT1dORVJfVT48T1dORV
JfTT4xPC9PV05FUl9NPjxPV05FUl9BPjA8L09XTkVSX0E+PEdST1VQX1U+MTwvR1JPVVBfVT48R1JPVVBfTT4xPC9HUk9VUF9NPjx
HUk9VUF9BPjA8L0dST1VQX0E+PE9USEVSX1U+MTwvT1RIRVJfVT48T1RIRVJfTT4xPC9PVEhFUl9NPjxPVEhFUl9BPjA8L09USEVS
X0E+PC9QRVJNSVNTSU9OUz48Q0xVU1RFUl9JRD4tMTwvQ0xVU1RFUl9JRD48Q0xVU1RFUj48L0NMVVNURVI+PFRZUEU+MDwvVFlQR
T48QlJJREdFPmJycDA8L0JSSURHRT48VkxBTj4wPC9WTEFOPjxQSFlERVY+ZW0yPC9QSFlERVY+PFZMQU5fSUQvPjxHTE9CQUxfUF
JFRklYLz48U0lURV9QUkVGSVgvPjxSQU5HRT48SVBfU1RBUlQ+MjEyLjIzMS41LjMzPC9JUF9TVEFSVD48SVBfRU5EPjIxMi4yMzE
uNS42MTwvSVBfRU5EPjwvUkFOR0U+PFRPVEFMX0xFQVNFUz4yPC9UT1RBTF9MRUFTRVM+PFRFTVBMQVRFPjxETlM+PCFbQ0RBVEFb
OC44LjguOCA4LjguNC40XV0+PC9ETlM+PEdBVEVXQVk+PCFbQ0RBVEFbMjEyLjIzMS41LjYyXV0+PC9HQVRFV0FZPjxORVRXT1JLX
0FERFJFU1M+PCFbQ0RBVEFbMjEyLjIzMS41LjMyLzI3XV0+PC9ORVRXT1JLX0FERFJFU1M+PE5FVFdPUktfTUFTSz48IVtDREFUQV
syNTUuMjU1LjI1NS4yMjRdXT48L05FVFdPUktfTUFTSz48L1RFTVBMQVRFPjxMRUFTRVM+PExFQVNFPjxNQUM+MDI6MDA6ZDQ6ZTc
6MDU6MjI8L01BQz48SVA+MjEyLjIzMS41LjM0PC9JUD48SVA2X0xJTks+ZmU4MDo6NDAwOmQ0ZmY6ZmVlNzo1MjI8L0lQNl9MSU5L
PjxVU0VEPjE8L1VTRUQ+PFZJRD4yMTU8L1ZJRD48L0xFQVNFPjxMRUFTRT48TUFDPjAyOjAwOmQ0OmU3OjA1OjIzPC9NQUM+PElQP
jIxMi4yMzEuNS4zNTwvSVA+PElQNl9MSU5LPmZlODA6OjQwMDpkNGZmOmZlZTc6NTIzPC9JUDZfTElOSz48VVNFRD4xPC9VU0VEPj
xWSUQ+MTk2NDwvVklEPjwvTEVBU0U+PC9MRUFTRVM+PC9WTkVUPg==",
  RADVD="NO",
  SSH_PUBLIC_KEY="ssh-dss
AAAAB3NzaC1kc3MAAACBANvNwJzq7l1voyfwn6P7X5MWq8tOXC8Uto3DXe7k+PZFsEfplxf7RhSo+/0eSfh1LqilmKZ+AUjWUk38t
IKE9rA0GXsr6xUptajy9SlaoELcBpAZgE0yHNUlzsB6Gm7qA3rG+Yk+WYXbJIdAczHYaD0cnPN0J+6l2R+xSYK0xt/RAAAAFQCq83
t+TjGmeUnGMoF7shAKLi2roQAAAIEAjktgsAgf/UzuCBLno3LNfQgslN28L7EmEN7YdtBjquGWxWPskta+e9A2m26ksX7L6td/PJS
rxDuDwFVWAE+tbDO8TOz8n5BGtSAt5VP3/vltK1950Kj4dlhZOD6WbTLnVHqBQFKOUCroeprEdMgDf/dpV2UYoCU0A5acLL3c5OIA
AACAfePn7/qO/fP1p4P98MxMZPhM5kfXbx3mEnt7EIsoLcRdWLpMZPhcnLvBV478n19Z4B87wX7wdo36JfcsDaUl/gKddytjfRN8/
xscgg/fMofFAjuvxIQcyomQ/HzFX4U7pSMEmmFiemgDvcuTsMN6g2EBjpBl/bZ1sIpK20/q9iE= oneadmin@one-node1",
  TARGET="hdb",

TEMPLATE="PFZNPjxJRD4xOTY0PC9JRD48VUlEPjA8L1VJRD48R0lEPjA8L0dJRD48VU5BTUU+b25lYWRtaW48L1VOQU1FPjxHTkF
NRT5vbmVhZG1pbjwvR05BTUU+PE5BTUU+Um91dGVyVkxBTjU8L05BTUU+PFBFUk1JU1NJT05TPjxPV05FUl9VPjE8L09XTkVSX1U+
PE9XTkVSX00+MTwvT1dORVJfTT48T1dORVJfQT4wPC9PV05FUl9BPjxHUk9VUF9VPjA8L0dST1VQX1U+PEdST1VQX00+MDwvR1JPV
VBfTT48R1JPVVBfQT4wPC9HUk9VUF9BPjxPVEhFUl9VPjA8L09USEVSX1U+PE9USEVSX00+MDwvT1RIRVJfTT48T1RIRVJfQT4wPC
9PVEhFUl9BPjwvUEVSTUlTU0lPTlM+PExBU1RfUE9MTD4wPC9MQVNUX1BPTEw+PFNUQVRFPjE8L1NUQVRFPjxMQ01fU1RBVEU+MDw
vTENNX1NUQVRFPjxSRVNDSEVEPjA8L1JFU0NIRUQ+PFNUSU1FPjEzODgwNzY3MTQ8L1NUSU1FPjxFVElNRT4wPC9FVElNRT48REVQ
TE9ZX0lEPjwvREVQTE9ZX0lEPjxNRU1PUlk+MDwvTUVNT1JZPjxDUFU+MDwvQ1BVPjxORVRfVFg+MDwvTkVUX1RYPjxORVRfUlg+M
DwvTkVUX1JYPjxURU1QTEFURT48Q1BVPjwhW0NEQVRBWzAuMl1dPjwvQ1BVPjxESVNLPjxDTE9ORT48IVtDREFUQVtZRVNdXT48L0
NMT05FPjxEQVRBU1RPUkU+PCFbQ0RBVEFbcWNvdzJfaW1hZ2VzXV0+PC9EQVRBU1RPUkU+PERBVEFTVE9SRV9JRD48IVtDREFUQVs
xMDBdXT48L0RBVEFTVE9SRV9JRD48REVWX1BSRUZJWD48IVtDREFUQVtoZF1dPjwvREVWX1BSRUZJWD48RElTS19JRD48IVtDREFU
QVswXV0+PC9ESVNLX0lEPjxEUklWRVI+PCFbQ0RBVEFbcWNvdzJdXT48L0RSSVZFUj48SU1BR0U+PCFbQ0RBVEFbdmlydHVhbF9yb
3V0ZXIgcWNvdzIgbm9uIHBlcnNpc3RlbnRdXT48L0lNQUdFPjxJTUFHRV9JRD48IVtDREFUQVs4XV0+PC9JTUFHRV9JRD48UkVBRE
9OTFk+PCFbQ0RBVEFbTk9dXT48L1JFQURPTkxZPjxTQVZFPjwhW0NEQVRBW05PXV0+PC9TQVZFPjxTT1VSQ0U+PCFbQ0RBVEFbL3Z
hci9saWIvb25lL2RhdGFzdG9yZXMvMTAwL2NkZWEzMDIwMjliNzI1ZTc0ZWQyZTY5MTY2ODU4ZDExXV0+PC9TT1VSQ0U+PFRBUkdF
VD48IVtDREFUQVtoZGFdXT48L1RBUkdFVD48VE1fTUFEPjwhW0NEQVRBW3Fjb3cyXV0+PC9UTV9NQUQ+PFRZUEU+PCFbQ0RBVEFbR
klMRV1dPjwvVFlQRT48L0RJU0s+PE1FTU9SWT48IVtDREFUQVsxMjhdXT48L01FTU9SWT48TklDPjxCUklER0U+PCFbQ0RBVEFbYn
JwMF1dPjwvQlJJREdFPjxJUD48IVtDREFUQVsyMTIuMjMxLjUuMzVdXT48L0lQPjxJUDZfTElOSz48IVtDREFUQVtmZTgwOjo0MDA
6ZDRmZjpmZWU3OjUyM11dPjwvSVA2X0xJTks+PE1BQz48IVtDREFUQVswMjowMDpkNDplNzowNToyM11dPjwvTUFDPjxORVRXT1JL
PjwhW0NEQVRBW0ludGVybmV0XV0+PC9ORVRXT1JLPjxORVRXT1JLX0lEPjwhW0NEQVRBWzJdXT48L05FVFdPUktfSUQ+PE5JQ19JR
D48IVtDREFUQVswXV0+PC9OSUNfSUQ+PFBIWURFVj48IVtDREFUQVtlbTJdXT48L1BIWURFVj48VkxBTj48IVtDREFUQVtOT11dPj
wvVkxBTj48L05JQz48TklDPjxCUklER0U+PCFbQ0RBVEFbYnJ2bDVdXT48L0JSSURHRT48SVA+PCFbQ0RBVEFbMTkyLjE2OC4xMjA
uMV1dPjwvSVA+PElQNl9MSU5LPjwhW0NEQVRBW2ZlODA6OjQwMDpjMGZmOmZlYTg6NzgwMV1dPjwvSVA2X0xJTks+PE1BQz48IVtD
REFUQVswMjowMDpjMDphODo3ODowMV1dPjwvTUFDPjxORVRXT1JLPjwhW0NEQVRBW1JUVkxBTjVdXT48L05FVFdPUks+PE5FVFdPU
```

```
ktfSUQ+PCFbQ0RBVEFbOV1dPjwvTkVUV09SS19JRD48TklDX0lEPjwhW0NEQVRBWzFdXT48L05JQ19JRD48UEhZREVWPjwhW0NEQV
RBW2VtNF1dPjwvUEhZREVWPjxWTEFOPjwhW0NEQVRBW1lFU11dPjwvVkxBTj48VkxBTl9JRD48IVtDREFUQVs1XV0+PC9WTEFOX0l
EPjwvTklDPjxPUz48QVJDSD48IVtDREFUQVt4ODZfNjRdXT48L0FSQ0g+PEJPT1Q+PCFbQ0RBVEFbaGRdXT48L0JPT1Q+PC9PUz48
VEVNUExBVEVfSUQ+PCFbQ0RBVEFbN11dPjwvVEVNUExBVEVfSUQ+PFZNSUQ+PCFbQ0RBVEFbMTk2NF1dPjwvVk1JRD48L1RFTVBMQ
VRFPjxVU0VSX1RFTVBMQVRFPjxGRUFUVVJFUz48QUNQST48IVtDREFUQVt5ZXNdXT48L0FDUEk+PC9GRUFUVVJFUz48R1JBUEhJQ1
M+PExJU1RFTj48IVtDREFUQVswLjAuMC4wXV0+PC9MSVNURU4+PFRZUEU+PCFbQ0RBVEFbVk5DXV0+PC9UWVBFPjwvR1JBUEhJQ1M
+PC9VU0VSX1RFTVBMQVRFPjxISVNUT1JZX1JFQ09SRFMvPjwvVk0+" ]
CPU="0.2"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  PORT="7864",
  TYPE="VNC" ]
MEMORY="128"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="7"
VMID="1964"
```

Bien, vamos replicando el resto de plantillas para adaptar las máquinas a las redes del nuevo usuario.

```
[oneadmin@one-node1 mytemplates]$ cp centos65VL2_3_contextNP.tmpl centos65VL5_6_contextNP.tmpl
[oneadmin@one-node1 mytemplates]$ vi centos65VL5_6_contextNP.tmpl
[oneadmin@one-node1 mytemplates]$ more centos65VL5_6_contextNP.tmpl
NAME="Centos65_x86_64 VLANs 5 6 NP"
CPU="0.5"
DISK=[
  IMAGE_ID="5",
  DEV_PREFIX="vd",
  TARGET="vda" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
FEATURES=[
  ACPI="yes" ]
NIC = [
  NETWORK="VLAN5",
  MODEL="virtio" ]
NIC = [
  NETWORK="VLAN6",
  MODEL="virtio" ]
```

```
CONTEXT=[
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
[oneadmin@one-node1 mytemplates]$ onetemplate create centos65VL5_6_contextNP.tmpl
ID: 8
[oneadmin@one-node1 mytemplates]$ onetemplate chown 8 3 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 8
TEMPLATE 8 INFORMATION
ID              : 8
NAME            : Centos65_x86_64 VLANs 5 6 NP
USER            : customer2
GROUP           : users
REGISTER TIME   : 12/26 18:01:47

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.5"
DISK=[
  DEV_PREFIX="vd",
  IMAGE_ID="5",
  TARGET="vda" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
NIC=[
  MODEL="virtio",
  NETWORK="VLAN5" ]
NIC=[
  MODEL="virtio",
  NETWORK="VLAN6" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
[oneadmin@one-node1 mytemplates]$ cp ttyVL2_3_contextNP.tmpl ttyVL5_6_contextNP.tmpl
[oneadmin@one-node1 mytemplates]$ more ttyVL5_6_contextNP.tmpl
```

```
NAME="tty_linuxV3 VLANs 5 6 NP"
CPU="0.1"
DISK=[
  IMAGE_ID="12" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
FEATURES=[
  ACPI="no" ]
NIC = [ NETWORK="VLAN5" ]
NIC = [ NETWORK="VLAN6" ]

CONTEXT=[
  NETWORK        = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
[oneadmin@one-node1 mytemplates]$ onetemplate chown 9 3 1
[oneadmin@one-node1 mytemplates]$ onetemplate show 9
TEMPLATE 9 INFORMATION
ID              : 9
NAME            : tty_linuxV3 VLANs 5 6 NP
USER            : customer2
GROUP           : users
REGISTER TIME   : 12/26 18:05:02

PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.1"
DISK=[
  IMAGE_ID="12" ]
FEATURES=[
  ACPI="no" ]
GRAPHICS=[
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="64"
NIC=[
  NETWORK="VLAN5" ]
```

```
NIC=[
  NETWORK="VLAN6" ]
```

Ya lo tenemos todo mas o menos igual para ambos usuarios.

Para probarlo, tenemos que eliminar unas cuantas instancias creadas del usuario customer1 (para ser justos, dejaremos la mitad de la capacidad total).

```
[oneadmin@one-node1 mytemplates]$ onevm delete 1724..1844
[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME            GROUP      AUTH           VMS            MEMORY         CPU
   0 oneadmin        oneadmin   core             -                 -           -
   1 serveradmin     oneadmin   server_c         -                 -           -
   2 customer1       users      core     104 /   0   51.6G /      0M  51.6 / 0.0
   3 customer2       users      core             -                 -           -
```

Ahora lanzamos una instancia en nombre del usuario customer2:

```
[oneadmin@one-node1 mytemplates]$ onetemplate list
  ID USER            GROUP           NAME                                REGTIME
   1 oneadmin        oneadmin        RouterVLAN3                  12/19 16:21:49
   2 oneadmin        oneadmin        tty_linuxV2 non persistent   12/20 11:36:41
   3 customer1       users           tty_linuxV2 VLAN3 NP         12/22 23:15:56
   4 customer1       users           Centos65_x86_64 VLAN3 NP     12/23 01:35:05
   5 customer1       users           tty_linuxV3 VLANs 2 3 NP     12/23 15:12:44
   6 customer1       users           Centos65_x86_64 VLANs 2 3 N  12/23 15:48:42
   7 oneadmin        oneadmin        RouterVLAN5                  12/26 17:49:54
   8 customer2       users           Centos65_x86_64 VLANs 5 6 N  12/26 18:01:47
   9 customer2       users           tty_linuxV3 VLANs 5 6 NP     12/26 18:05:02
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 8 --name "Centos65VL5_6_test" --user
customer2 --password 2solomio
VM ID: 1967
```

Esto funciona bien. Vamos a eliminar todas las máquinas en ejecución, para ambos usuarios:

```
[oneadmin@one-node1 one]$ onevm delete 1845..1946
```

Con eso dejamos la plataforma preparada para que los usuarios puedan invocar sus máquinas virtuales.

6. Asignación de cuotas de uso a los usuarios

Ahora tenemos que trabajar con las cuotas de uso para cada usuario.

Actualmente, cada usuario podría invocar recursos hasta agotar los recursos físicos proporcionados por ambos hosts. Para evitar la canibalización de recursos por parte de los usuarios en detrimento del resto, vamos a definir los ficheros de cuotas para aplicar a cada uno. Para ello tenemos que definir la variable EDITOR en la cuenta oneadmin:

```
[oneadmin@one-node1 mytemplates]$ oneuser quota customer1
Editor not defined
[oneadmin@one-node1 mytemplates]$ type vim
vim is hashed (/usr/bin/vim)
```

```
[oneadmin@one-node1 mytemplates]$ echo $EDITOR

[oneadmin@one-node1 mytemplates]$ export EDITOR="/usr/bin/vim"
[oneadmin@one-node1 mytemplates]$ echo $EDITOR
/usr/bin/vim
[oneadmin@one-node1 mytemplates]$ oneuser quota customer1
```

Y ahora si que nos deja editar las cuotas. Se invoca el editor vim y nos aparece esto:

```
#-----------------------------------------------------------------------
# Supported quota limits:
#
#  DATASTORE = [
#    ID     = <ID of the datastore>
#    IMAGES = <Max. number of images in the datastore>
#    SIZE   = <Max. storage capacity (Mb) used in the datastore>
#  ]
#
#  VM = [
#    VMS    = <Max. number of VMs>
#    MEMORY = <Max. allocated memory (Mb)>
#    CPU    = <Max. allocated CPU>
#  ]
#
#  NETWORK = [
#    ID     = <ID of the network>
#    LEASES = <Max. number of IP leases from the network>
#  ]
#
#  IMAGE = [
#    ID     = <ID of the image>
#    RVMS   = <Max. number of VMs using the image>
#  ]
#
#  In any quota:
#    -1 means use the default limit ('defaultquota' command)
#    0 means unlimited.
#
#  The usage counters "*_USED" are shown for information
#  purposes and will NOT be modified.
#-----------------------------------------------------------------------
```

Mejor imposible. Vamos a fijar limite principalmente sobre el datastore raw, porque es el que más espacio consume, y sobre todo en los valores dentro de las VMs, ya que el usuario no tiene acceso a la lan pública.

Estos serán los valores:

```
DATASTORE 1 (default, raw): 200G (200 * 1024M) y sin límite de imágenes.
VMs = 100
MEMORY = 50G (50 * 1024M)
CPU = 50
NETWORK 2 (internet): 3 IPs máximo. No cuentan las IPs de los routers virtuales, porque pertenecen al administrador.
```

Con estos datos, dejamos definidas las mismas cuotas para ambos usuarios:

```
#-----------------------------------------------------------------------
# Supported quota limits:
#
DATASTORE = [
    ID     =  1,
    IMAGES =  0,
    SIZE   =  204800
]

VM = [
    VMS    = 100,
    MEMORY = 51200,
    CPU    = 50
]

NETWORK = [
    ID     = 2,
    LEASES = 3
]

#  IMAGE = [
#    ID     = <ID of the image>
#    RVMS   = <Max. number of VMs using the image>
#  ]
#
#  In any quota:
#    -1 means use the default limit ('defaultquota' command)
#    0 means unlimited.
#
#  The usage counters "*_USED" are shown for information
#  purposes and will NOT be modified.
#-----------------------------------------------------------------------


[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME            GROUP      AUTH           VMS            MEMORY         CPU
   0 oneadmin        oneadmin   core             -                 -           -
   1 serveradmin     oneadmin   server_c         -                 -           -
   2 customer1       users      core        0 / 100       0M /     50G  0.0 / 50.0
   3 customer2       users      core                 -                 -           -
[oneadmin@one-node1 mytemplates]$ oneuser show 2
USER 2 INFORMATION
ID             : 2
NAME           : customer1
GROUP          : users
PASSWORD       : 9e8ce8b24a55028eb4cad409375f107a4700a781
AUTH_DRIVER    : core
ENABLED        : Yes

USER TEMPLATE
TOKEN_PASSWORD="67e9aedafe4fc2280283d96364bbc0561a034490"

RESOURCE USAGE & QUOTAS


       NUMBER OF VMS               MEMORY                  CPU
        0 /      100        0M /      50G      0.00 /     50.00


DATASTORE ID               IMAGES                SIZE
           1         0 /        0       0M /      200G


  NETWORK ID               LEASES
           2         0 /        3
```

Ahora hacemos lo mismo con el otro usuario:

```
[oneadmin@one-node1 mytemplates]$ oneuser quota customer2
[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME            GROUP      AUTH           VMS            MEMORY         CPU
   0 oneadmin        oneadmin   core                 -                 -           -
   1 serveradmin     oneadmin   server_c             -                 -           -
   2 customer1       users      core        0 / 100       0M /     50G  0.0 / 50.0
   3 customer2       users      core        0 / 100       0M /     50G  0.0 / 50.0
```

Sólo nos queda definir las cuotas por defecto para los siguientes usuarios:

```
[oneadmin@one-node1 mytemplates]$ oneuser defaultquota
```

Y definimos los siguientes valores por defecto para cualquier usuario que no tenga definida ninguna cuota explícitamente.

```
DATASTORE = [
    ID     =  1,
    IMAGES =  0,
    SIZE   =  102400
]
```

```
VM = [
    VMS    = 30,
    MEMORY = 15360,
    CPU    = 15
]

NETWORK = [
    ID     = 2,
    LEASES = 1
]
```

También podemos definir las cuotas para todos los usuarios de un mismo grupo, en esta caso podríamos haberlo hecho con el grupo "users".

7. Últimos pasos de configuración de la plataforma

Para terminar, dejaremos levantadas las instancias de los routers:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 1 --name "RouterVLAN3"
VM ID: 1968
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 7 --name "RouterVLAN5"
VM ID: 1969
[oneadmin@one-node1 mytemplates]$ onevm  list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
  1968 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 00h03
  1969 oneadmin oneadmin RouterVLAN5     runn    0    128M one-node1    0d 00h01
[oneadmin@one-node1 mytemplates]$ onevnet list
  ID USER         GROUP        NAME            CLUSTER      TYPE BRIDGE   LEASES
   1 customer1    users        VLAN2           -               R brvl2         0
   2 oneadmin     oneadmin     Internet        -               R brp0          2
   5 customer1    users        VLAN3           -               R brvl3         1
   6 oneadmin     oneadmin     RTVLAN3         -               R brvl3         1
   7 customer2    users        VLAN6           -               R brvl6         0
   8 customer2    users        VLAN5           -               R brvl5         1
   9 oneadmin     oneadmin     RTVLAN5         -               R brvl5         1
[oneadmin@one-node1 mytemplates]$ onehost list
  ID NAME            CLUSTER   RVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   0 one-node1       -           1     20 / 8000 (0%)  128M / 62.9G (0%) on
   1 one-node2       -           1     20 / 8000 (0%)  128M / 62.9G (0%) on
```

Nos queda preparar una imagen que pueda conectarse directamente a internet con una IP pública. Para ello aprovecharemos las cuotas que acabamos de configurar sobre la red 2 ("Internet"), junto con el permiso de utilización asignado a esa red. La plantilla generada será la siguiente:

```
[oneadmin@one-node1 mytemplates]$ onetemplate clone 8 "Centos65_x86_64 Internet NP"
ID: 10
[oneadmin@one-node1 mytemplates]$ onetemplate update 10
```

Ahora vamos a darle permisos para que pueda ser utilizada en común por todos los usuarios del grupo users:

```
[oneadmin@one-node1 mytemplates]$ onetemplate chgrp 10 1
[oneadmin@one-node1 mytemplates]$ onetemplate chmod 10 660
[oneadmin@one-node1 mytemplates]$ onetemplate show 10
TEMPLATE 10 INFORMATION
ID             : 10
NAME           : Centos65_x86_64 Internet NP
USER           : oneadmin
GROUP          : users
REGISTER TIME  : 12/27 21:12:36

PERMISSIONS
OWNER          : um-
GROUP          : um-
OTHER          : ---

TEMPLATE CONTENTS
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]" ]
CPU="0.5"
DISK=[
  DEV_PREFIX="vd",
  IMAGE_ID="5",
  TARGET="vda" ]
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  KEYMAP="es",
  LISTEN="0.0.0.0",
  TYPE="VNC" ]
MEMORY="512"
NIC=[
  MODEL="virtio",
  NETWORK="Internet",
  NETWORK_UNAME="oneadmin" ]
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
```

Como mejoras hemos añadido el teclado en español para la interfaz de consola por VNC, y hemos indicado explícitamente que el propietario de la red pública es "oneadmin" (el administrador), para que los usuarios del grupo "users", no tengan problemas en utilizar una red de la que no son titulares, si tienen derecho de uso y cuota. Con esto resolvemos también el problema que habíamos detectado con la instancia del router virtual.

Esta plantilla, tal cual está, es utilizable por todos los usuarios del grupo “users”, incluyendo a customer1 y customer2.

Vamos a probarlo:

```
[oneadmin@one-node1 mytemplates]$ onetemplate instantiate 10 --name "CentOS65-pub1" --user customer1
--password "1solomio"
VM ID: 1973
[oneadmin@one-node1 mytemplates]$ onevm list
    ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
  1968 oneadmin oneadmin RouterVLAN3     runn    0    128M one-node2    0d 21h49
  1970 oneadmin oneadmin Centos65VL5_6_1 runn    0  661.9M one-node1    0d 20h23
  1971 oneadmin oneadmin RouterVLAN5     runn    0    128M one-node1    0d 19h55
  1972 oneadmin oneadmin Centos65VL2_3_1 runn    0  669.3M one-node2    0d 19h37
  1973 customer users    CentOS65-pub1   runn   88    512M one-node2    0d 00h00
[oneadmin@one-node1 mytemplates]$ onevm show 1973
VIRTUAL MACHINE 1973 INFORMATION
ID                  : 1973
NAME                : CentOS65-pub1
USER                : customer1
GROUP               : users
STATE               : ACTIVE
LCM_STATE           : RUNNING
RESCHED             : No
HOST                : one-node2
START TIME          : 12/27 21:24:16
END TIME            : -
DEPLOY ID           : one-1973

VIRTUAL MACHINE MONITORING
NET_RX              : 3K
NET_TX              : 1K
USED MEMORY         : 512M
USED CPU            : 0

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---

VM DISKS
 ID TARGET IMAGE                              TYPE SAVE SAVE_AS
  0 vda    CentOS 6.5 non persistent          file  NO        -

VM NICS
 ID NETWORK              VLAN BRIDGE       IP              MAC
  0 Internet               no brp0         192.101.5.35    02:00:d4:e7:05:23
```

```
                                           fe80::400:d4ff:fee7:523

VIRTUAL MACHINE HISTORY
SEQ HOST            ACTION           REAS           START         TIME     PROLOG
  0 one-node2       none             none  12/27 21:24:27   0d 00h03m   0h00m00s

VIRTUAL MACHINE TEMPLATE
CONTEXT=[
  DISK_ID="1",
  ETH0_DNS="8.8.8.8 8.8.4.4",
  ETH0_GATEWAY="192.101.5.62",
  ETH0_IP="192.101.5.35",
  ETH0_MASK="255.255.255.224",
  ETH0_NETWORK="192.101.5.32/27",
  NETWORK="YES",
  TARGET="hda" ]
CPU="0.5"
FEATURES=[
  ACPI="yes" ]
GRAPHICS=[
  KEYMAP="es",
  LISTEN="0.0.0.0",
  PORT="7873",
  TYPE="VNC" ]
MEMORY="512"
OS=[
  ARCH="x86_64",
  BOOT="hd" ]
TEMPLATE_ID="10"
VMID="1973"
```

Vemos que la configuración de red se ha asignado correctamente, y que la máquina es accesible desde internet:

```
[jamontes@Tesla ~]$ ssh root@192.101.5.35
The authenticity of host '192.101.5.35 (192.101.5.35)' can't be established.
RSA key fingerprint is 3f:d6:b0:75:21:0a:3e:93:53:5a:ee:8e:b9:8a:9e:17.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.101.5.35' (RSA) to the list of known hosts.
root@192.101.5.35's password:
Last login: Fri Dec 27 21:27:33 2013
[root@localhost ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:D4:E7:05:23
          inet addr:192.101.5.35  Bcast:192.101.5.63  Mask:255.255.255.224
          inet6 addr: fe80::d4ff:fee7:523/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:141 errors:0 dropped:0 overruns:0 frame:0
          TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16697 (16.3 KiB)  TX bytes:10013 (9.7 KiB)

[root@localhost ~]# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
[root@localhost ~]# iptables-save
# Generated by iptables-save v1.4.7 on Fri Dec 27 21:31:43 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [103:12279]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Fri Dec 27 21:31:43 2013
[root@localhost ~]# who
root     tty1         2013-12-27 21:27
root     pts/0        2013-12-27 21:30 (adijon-151-1-63-245.w83-196.abo.wanadoo.fr)
```

Si vemos los datos del usuario customer1, veremos como va consumiendo la cuota de uso de la red pública:

```
[oneadmin@one-node1 mytemplates]$ oneuser list
  ID NAME            GROUP      AUTH           VMS            MEMORY         CPU
   0 oneadmin        oneadmin   core             -                 -           -
   1 serveradmin     oneadmin   server_c         -                 -           -
   2 customer1       users      core       1 / 100    512M /      50G  0.5 / 50.0
   3 customer2       users      core       0 / 100      0M /      50G  0.0 / 50.0
[oneadmin@one-node1 mytemplates]$ oneuser show 2
USER 2 INFORMATION
ID             : 2
NAME           : customer1
GROUP          : users
PASSWORD       : 9e8ce8b24a55028eb4cad409375f107a4700a781
AUTH_DRIVER    : core
ENABLED        : Yes

USER TEMPLATE
TOKEN_PASSWORD="67e9aedafe4fc2280283d96364bbc0561a034490"
```

```
RESOURCE USAGE & QUOTAS


       NUMBER OF VMS               MEMORY                  CPU
        1 /      100      512M /      50G      0.50 /     50.00

DATASTORE ID               IMAGES                SIZE
           1         0 /        0      0M /      200G

  NETWORK ID               LEASES
           2         1 /        3

    IMAGE ID          RUNNING VMS
           5         1 /        0
```

Con estas pruebas concluimos la fase de preproducción. La plataforma está preparada para el servicio.