

Instalacion, despliegue y gestión de OpenNebula

José Manuel Benítez Sánchez

30 de marzo de 2015



Índice

1. Introducción	4
2. Características, arquitectura y conceptos	4
2.1. Arquitectura	6
2.2. Front-End	6
2.3. Monitorización	7
2.4. Virtualización de Hosts	7
2.5. Redes	7
2.6. Almacenamiento	8
2.7. Autenticación	9
2.8. Otros componentes avanzados	9
3. Instalación para linux CentOS 7	9
3.1. Requisitos previos	9
3.1.1. Deshabilitar SELINUX	10
3.1.2. Deshabilitar Firewall	10
3.2. Instalando el Front-End en Servidor	10
3.2.1. Instalando el repositorio	10
3.2.2. Instalamos los paquetes necesarios	10
3.2.3. Configurar y lanzar los servicios	11
3.2.4. Configuración de NFS	11
3.2.5. Configuración de la llave pública para SSH	11
3.3. Instalación en los Nodos	11
3.3.1. Añadir los repositorios	12
3.3.2. Instalación de los paquetes necesarios	12
3.3.3. Configuración de la Red	12
3.3.4. Configuración de NFS	13
3.4. Pasos finales y verificaciones	13
4. Despliegue y gestión de OpenNebula	13
4.1. Registrar nodos en OpenNebula	13
4.2. Añadir un Interfaz de Red Virtual	14
4.3. Usuario oneadmin	14
4.4. Gestión de OpenNebula a través del entorno web	15
4.4.1. Interfaz de Web, panel de control	15
4.5. MaketPlace	17
4.6. Despliegue de la máquina virtual	18
4.6.1. Creación de máquina virtual	19
4.6.2. Acceso a la máquina virtual	20
5. Despliegue y gestión de OpenNebula utilizando la línea de comandos	21
5.1. Registro de nodos	21
5.2. Añadir un Interfaz de Red Virtual	21
5.3. Usuario oneadmin	22
5.4. Descargar una imagen del MarketPlace	22
5.5. Creación de plantilla para la imagen descargada	22
5.6. Ejecución de la Máquina Virtual	23
5.7. Consulta de la lista de máquinas virtuales	23
5.8. Lista de comando útiles	23

1. Introducción

OpenNebula es una herramienta de software Open Source que cubre el modelo de servicio *IaaS*.

OpenNebula ofrece una solución sencilla pero con el añadido de tener múltiples características y de ser flexible para construir y gestionar nubes empresariales y centros de datos virtualizados.

OpenNebula está diseñado para ser simple y es fácil de instalar, actualizar y operar por los administradores, y sencillo de utilizar por los usuarios finales. Al estar centrado en la simplicidad, se integra con las tecnologías existentes siempre que sea posible. OpenNebula trabaja con *MySQL*, *Ceph*, *LVM*, *GlusterFS*, *Open vSwitch*, *Ceph*, *LDAP* ...

OpenNebula es una solución software libre (bajo licencia *Apache v2*) que permite implementar fácilmente infraestructuras *Cloud Computing* privadas (también híbridas) según el modelo *IaaS*. Su parte principal consiste en software que permite desplegar máquinas virtuales sobre un *pool* de máquinas físicas. Además está diseñado para integrarse con otras soluciones de almacenamiento y de red. Así, maneja tanto la transferencia de máquinas virtuales como la configuración de la red, el almacenamiento y su gestión.

OpenNebula fue inicialmente desarrollado por la Universidad Complutense de Madrid en 2008. Más adelante el número de participantes ha ido creciendo y más organizaciones se han unido al desarrollo del proyecto. Algunos destacados contribuidores son *IBM*, *Suse* o *AT&T*

2. Características, arquitectura y conceptos

Almacenamiento Permite almacenar las imágenes de discos virtuales en repositorios desde donde serán usadas para desplegar rápidamente máquinas virtuales o compartidas con otros usuarios. Estas imágenes de disco pueden ser tanto de Sistemas Operativos o de datos.

Repositorio de plantillas Es donde se almacenan plantillas de máquinas virtuales, con sus características para ser instanciadas más tarde en el hipervisor elegido.

Redes virtuales Soporta el manejo de redes virtuales que interconectará las diferentes máquinas virtuales, pudiendo definir una IP concreta o rangos de IP para cada red.

Manejo de máquinas virtuales Una vez que se ha desplegado una instancia de una plantilla en un hipervisor (host) se puede controlar todo su ciclo de vida como el arranque, parada, clonación y apagado.

Clústeres Son *pools* de *host* que comparten almacenamiento y redes virtuales. Se utilizan para el balanceo de carga y alta disponibilidad y rendimiento.

Usuarios y grupos Soporta la definición de usuarios y grupos para el acceso a los recursos así como mecanismo de autenticación. También implementa *ACL* para la asignación permisos.

API Proporciona interfaces de comunicación con las diferentes funcionalidades ofrecidas a través desde herramientas de línea de comando o través del *GUI Web Sunstone*. Además proporciona interfaces para interactuar con otras infraestructuras *Cloud Computing* públicas como *OCCI7* y *EC28* con lo que permite el despliegue de nubes híbridas.

Multiples hipervisores Destaca principalmente por su flexibilidad para adaptarse a diferentes escenarios. En parte, esta flexibilidad la consigue gracias a que permite la utilización de diferentes soluciones de virtualización o hipervisores como *Xen*, *KVM* o *VMWare*. Estas tecnologías de virtualización son el corazón de OpenNebula.

OpenNebula permite el control de varios tipos diferentes de hipervisores como hemos indicado arriba: Xen, KVM y VMWare, a través de sus respectivos drivers.

Cada hipervisor registrado en el sistema se le denomina host o anfitrión. Así podemos tener varios host (máquinas físicas) y dentro de estas correr varias máquinas virtuales. Una máquina virtual en OpenNebula consiste de las siguientes partes:

- Capacidad de procesamiento en términos de uso de CPU.
- Un conjunto de dispositivos de red asociados a una red virtual.
- Un conjunto de imágenes de disco.
- Un fichero de recuperación donde se almacena la memoria de la máquina virtual que se está ejecutando e información del hipervisor.

Debido a que no hay dos nubes iguales, OpenNebula ofrece muchas interfaces diferentes que se pueden utilizar para interactuar con la funcionalidad ofrecida para administrar los recursos físicos y virtuales. Hay cuatro principales perspectivas diferentes para interactuar con OpenNebula:

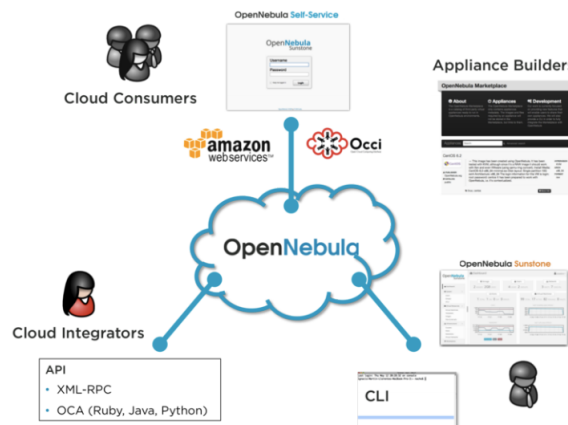


Figura 1: Interfaces de OpenNebula

- Interfaces de nubes para los consumidores, como la nube *EC2* de consulta e interfaces de *EBS*.
- Interfaces de administración para usuarios de la nube avanzadas y operadores, como una interfaz de línea de comandos de *Unix* y el poderoso *Sunstone GUI*.
- APIs de bajo nivel para la nube, Integradores en Ruby, Java y API XMLRPC
- Un mercado para 'builders' con un catálogo de dispositivos virtuales listos para funcionar en entornos OpenNebula.

2.1. Arquitectura

OpenNebula asume que su infraestructura física adopta una arquitectura de racimo clásico con un Front-End, y un conjunto de *hosts* donde las máquinas virtuales (VM)

Los componentes básicos de un sistema de OpenNebula son:

- Front-End que ejecuta los servicios OpenNebula.
- *Hosts* que proporcionan los recursos necesarios para las máquinas.
- *Datastores* que soportan las imágenes base de las máquinas virtuales.
- Redes físicas utilizadas para apoyar los servicios básicos tales como la interconexión de los servidores de almacenamiento y operaciones de control OpenNebula y VLAN para las máquinas virtuales.

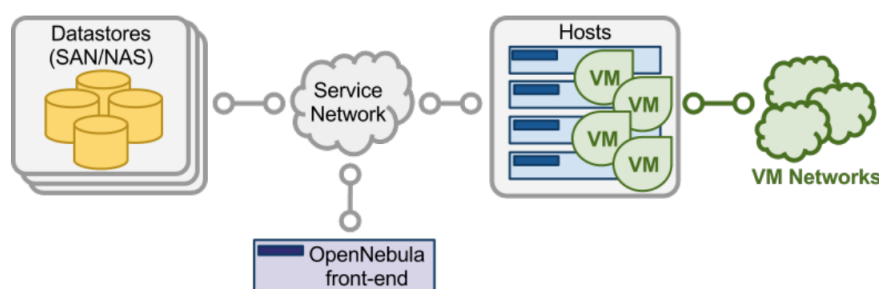


Figura 2: Componentes básicos de OpenNebula

OpenNebula presenta una arquitectura altamente modular que ofrece un amplio apoyo a los productos básicos y de nivel empresarial, servicios de hipervisor, monitoreo, almacenamiento, redes y gestión de usuarios.

2.2. Front-End

La máquina que mantiene la instalación OpenNebula se llama el FrontEnd. Esta máquina tiene conectividad de red para cada huésped, y, posiblemente, el acceso a los almacenes de datos de almacenamiento (ya sea por montaje directo o de red). La instalación de la base de OpenNebula ocupa menos de 50 MB. OpenNebula servicios incluyen:

- **Daemon** de gestión (oned) y planificador (mm_sched)
- Servidor de interfaz web (SunStone WebServe)

Hay varias plataformas certificadas para actuar como Front-End para cada versión de OpenNebula. La base de datos predeterminada de OpenNebula utiliza SQLite aunque también admite MySQL.

El número máximo de servidores (hosts de virtualización) que pueden ser manejados por una sola instancia OpenNebula (zona) depende en gran medida el rendimiento y la escalabilidad de la infraestructura de plataforma subyacente, principalmente el subsistema de almacenamiento.

No se recomienda más de 500 servidores dentro de cada zona, pero hay usuarios con 1.000 servidores en cada zona.

2.3. Monitorización

El subsistema de supervisión y monitorización recopila información relativa a los anfitriones y las máquinas virtuales, tales como el estado anfitrión, indicadores de eficiencia básicos, así como estado de MVs y capacidad de consumo. Esta información se recoge mediante la ejecución un conjunto de sondeos proporcionados por OpenNebula. La salida de estas sondas se envía a OpenNebula en dos diferentes maneras:

- **Modelo UDP-push** Cada host envía periódicamente los datos de seguimiento a través de UDP para la interfaz que lo recoge y lo procesa en un módulo dedicado. Este modelo es altamente escalable y su límite (en términos de número de máquinas virtuales monitoreado por segundo) se limita al rendimiento del servidor que ejecuta ONED y el servidor de base de datos.
- **Modelo de Sondeo:** OpenNebula periódicamente consultas activamente cada *host* y ejecuta las sondas a través de *ssh*. Este modo está limitado por el número de conexiones activas que puede hacerse al mismo tiempo, como anfitriones se consultan de forma secuencial.

2.4. Virtualización de Hosts

Los anfitriones son los equipos físicos que se ejecutarán las máquinas virtuales. Hay varias plataformas certificadas para que actúen como nodos para cada versión de OpenNebula. El subsistema de OpenNebula es el componente encargado de hablar con el hipervisor instalado en los hosts y tomando las acciones necesaria para cada paso en el ciclo de vida de MV.

OpenNebula soporta de forma nativa tres hipervisores:

- Xen
- KVM
- VMware

2.5. Redes

OpenNebula proporciona un subsistema de red fácilmente adaptable y personalizable a fin de integrar mejor los requisitos específicos de la red de centros de datos existentes. Se necesitan al menos dos redes físicas diferentes:

- Una red de servicio es necesaria por los servicios de Front-End OpenNebula para acceder a los anfitriones con el fin de gestionar y monitorizar a los hipervisores, y mover los archivos de imágenes.
- Se necesita una red instancia para ofrecer conectividad de red a los fabricantes a través de las diferentes máquinas.

El administrador OpenNebula puede asociar uno de los siguientes controladores a cada *host*:

- **dummy:** Controlador predeterminado que no realiza ninguna operación de la red. También se tienen en cuenta las reglas del cortafuego.
- **FW:** Las reglas de *firewall* se aplican, pero el aislamiento de redes se ignora.

- **802.1Q**: restringir el acceso a la red a través de etiquetado VLAN, que también requiere el apoyo del hardware.
- **ebtables**: restringir el acceso a la red a través de reglas *ebtables*. No es necesaria una especial configuración de hardware.
- **ovswitch**: restringir el acceso a la red con Open vSwitch Virtual Switch.
- **VMware**: utiliza la infraestructura de red de VMware para proporcionar una red compatible aislada.

2.6. Almacenamiento

OpenNebula utiliza almacenes de datos para manejar las imágenes de disco de máquina virtual. Un almacén de datos es cualquier medio de almacenamiento utilizado para almacenar disco e imágenes de máquinas virtuales. Tipicamente, un almacén de datos será respaldado por los servidores SAN / NAS. En general, cada almacén de datos tiene que ser accesible a través del Front-End utilizando cualquier tecnología adecuada NAS, SAN o almacenamiento de conexión directa.

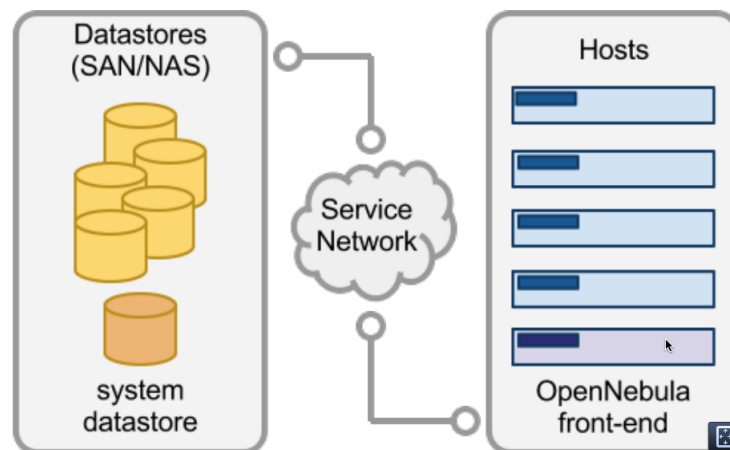


Figura 3: Componentes básicos de OpenNebula

Cuando se implementa una máquina virtual las imágenes se transfieren desde el almacén de datos para los anfitriones. Dependiendo del almacenamiento real la tecnología utilizada puede significar una transferencia real, un enlace simbólico o la creación de un volumen LVM.

OpenNebula se entrega con 3 clases del almacén de datos diferentes:

- **Datastores** es el sistema que almacenará las imágenes de máquinas virtuales en ejecución, en función de la tecnología de almacenamiento utilizado estas imágenes pueden ser copias completas de la imagen original, *qcow* deltas o enlaces de sistemas de archivos simples.
- *Datastores Image Store* es el repositorio de imágenes de disco. Las imágenes de disco se mueven, o clonan, etc. desde el sistema almacén de datos cuando las máquinas virtuales se despliegan.
- El almacén de datos de archivos es un almacén de datos especial que se utiliza para almacenar archivos sin formato y no imágenes de disco. Los archivos planos se pueden utilizar como *ramdisks* o archivos de contexto. Los almacenes de datos de imágenes pueden ser de diferente tipo en función de la tecnología:

- **vmfs**, un almacén de datos en formato VMFS para ser utilizado con los hipervisores VMware.
- **LVM**, el controlador del almacén de datos LVM ofrece la posibilidad de utilizar volúmenes LVM en lugar de archivos planos que contienen las imágenes virtuales.
- **Ceph**, para almacenar imágenes de disco utilizando dispositivos de bloque *Ceph*.

2.7. Autenticación

Se pueden elegir varios métodos de autenticación para acceder a OpenNebula:

- Usuario/Clave
- SSH Auth
- X509 Auth
- LDAP Auth

2.8. Otros componentes avanzados

Una vez que tenga una nube OpenNebula en marcha y funcionando, puede instalar los siguientes componentes avanzados:

- Aplicaciones Multi-VM y auto-escala: OneFlow que permite a los usuarios y administradores definir, ejecutar y gestionar aplicaciones de varios niveles, o servicios compuestos por máquinas virtuales interconectadas. Cada grupo de máquinas virtuales se implementa y se gestiona como una sola entidad, y es completamente integrado con la gestión de usuarios y grupo avanzado OpenNebula.
- Nube Bursting: es un modelo en el que los recursos locales de una nube privada se combinan con recursos de los proveedores remotos. Esta ayuda a y permite crear entornos de alojamiento altamente escalables.
- Nube pública: las interfaces de la nube se pueden agregar a su nube privada. Esta interfaz proporciona una sencilla y la gestión remota de los recursos (virtual) de la nube a un alto nivel de abstracción: *Amazon EC2* y *APIs* de *EBS*.
- Aplicación *Insight*: *OneGate* permite a los huéspedes de máquinas virtuales gestionar información de control a OpenNebula.

3. Instalación para linux CentOS 7

3.1. Requisitos previos

El propósito de esta guía es proporcionar a los usuarios paso a paso guía para instalar OpenNebula usando *CentOS 7* como sistema operativo y *KVM* como el hipervisor.

Después de seguir esta guía, los usuarios tendrán un OpenNebula trabajar con interfaz gráfica (Piedra del Sol), al menos un hipervisor (host) y unas máquinas virtuales en ejecución. Esto es útil en el momento de la creación de nubes piloto, para probar rápidamente nuevas características y despliegue como base para construir una gran infraestructura.

A lo largo de la instalación hay dos funciones separadas: Front-End y Nodos. El servidor Front-End ejecutará los servicios OpenNebula, y los nodos serán utilizados para ejecutar máquinas virtuales. Por favor, no es que sea posible seguir esta guía con sólo un anfitrión que combina tanto el Front-End y roles de los nodos de un solo servidor. Sin embargo, se recomienda ejecutar máquinas virtuales en hosts con extensiones de virtualización. Para probar si su host admite extensiones de virtualización, ejecute:

```
grep -E 'svm|vmx' /proc/cpuinfo
```

3.1.1. Deshabilitar SELINUX

Es necesario sólo para estas pruebas deshabilitar el servicio *SELinux*, para ello se debe usar un editor y modificar el fichero */etc/sysconfig/selinux*

```
SELINUX=disabled
```

Guardar y reiniciar el Equipo.

Una vez hecho esto, comprobar que *SELinux* está deshabilitado:

```
[root@master ~]# setenforce 0
[root@master ~]# getenforce
Permissive
```

3.1.2. Deshabilitar Firewall

Al igual que ocurre con el *SELinux* es necesario deshabilitar el Firewall de CentOS solo para la instalación de pruebas que vamos a realizar.

Hay que usar los siguientes comandos:

```
[root@master ~]# systemctl disable firewalld
[root@master ~]# systemctl stop firewalld
```

3.2. Instalando el Front-End en Servidor

Para la instalación usaremos dos perfiles de usuario: *root* y *oneadmin*

3.2.1. Instalando el repositorio

Ejecutar como root:

```
[root@master ~]# yum install epel-release
```

Añadimos el repositorio de OpenNebula:

Ejecutar como root:

```
[root@master ~]# cat << EOT > /etc/yum.repos.d/opennebula.repo
[opennebula]
name=opennebula
baseurl=http://downloads.opennebula.org/repo/4.8/CentOS/7/x86_64/
enabled=1
gpgcheck=0
EOT
```

3.2.2. Instalamos los paquetes necesarios

Instalamos los servicios de OpenNebula y OpenNebula SunStone:

Ejecutar como root:

```
[root@master ~]# yum install opennebula-server opennebula-sunstone
```

Ahora instalamos todas las dependencias de las paquetes de *Ruby*.

```
Ejecutar como root:

[root@master ~]# /usr/share/one/install_gems

La aplicacion te informa que tipo de Linux usas, seleccionar CentOS.
```

3.2.3. Configurar y lanzar los servicios

Hay dos procesos principales que se ejecutan para OpenNebula: **oned**, y el interfaz gráfico de usuario: **sunstone**.

```
Ejecutar como root:

[root@master ~]# service opennebula start
[root@master ~]# service opennebula-sunstone start
```

3.2.4. Configuración de NFS

Si vas a desplegar OpenNebula en dos o más nodos haz este paso. Si en caso contrario estás usando un sólo servidor tanto para el Front-End como para los nodos de trabajo pasa al siguiente paso.

Exporta con NFS: `/var/lib/one` para ello, edita el fichero `/etc/exports`

```
Include esta linea al fichero /etc/exports :

/var/lib/one/ *(rw,sync,no_subtree_check,root_squash)
```

Luego reinicia el servicio de NFS:

```
[root@master ~]# systemctl restart nfs.service
```

3.2.5. Configuración de la llave pública para SSH

Para utilizar OpenNebula se debe configurar *SSH* sin clave para que se pueda acceder desde cualquier nodo a otro nodo.

```
Ejecutar como usuario oneadmin

[root@master ~]# su - oneadmin
[oneadmin@master ~]# cat << EOT > ~/.ssh/config
Host *
    StrictHostKeyChecking no
    UserKnownHostsFile /dev/null
EOT
```

Cambiamos los permisos del fichero `/.ssh/config`

```
Ejecutar como usuario oneadmin

[oneadmin@master ~]# chmod 600 ~/.ssh/config
```

Con este último paso ya hemos terminado la primera parte de la instalación del servicio en el nodo principal. Ahora, pasaremos a instalar los demás servicios en los demás nodos.

3.3. Instalación en los Nodos

Esta parte de la instalación se hace para cada uno de los nodos que tengamos en nuestro conjunto de nodos de un clúster de máquinas.

Si habéis elegido usar la Instalación del **Front-End** y **Nodos** en la misma Máquina, igualmente debéis instalar estos servicios.

Importante: Hay que desactivar los servicios de *Firewall* y *SELinux*, al igual que se realizó en la sección de *??*. Siga estos pasos para desactivar *firewalld* (*iptables*) y *SELinux* y luego prosiga con la instalación siguiente.

Todos los pasos siguientes se tienen que realizar para cada nodo extra que queramos añadir a OpenNebula.

3.3.1. Añadir los repositorios

Añadimos el repositorio para instalar el software que correrá en los nodos de trabajo.

```
Ejecutar como usuario root

[root@node01 ~]# cat << EOT > /etc/yum.repos.d/opennebula.repo
[opennebula]
name=opennebula
baseurl=http://downloads.opennebula.org/repo/4.8/CentOS/7/x86_64/
enabled=1
gpgcheck=0
EOT
```

3.3.2. Instalación de los paquetes necesarios

```
Ejecutar como usuario root

[root@node01 ~]# yum install opennebula-node-kvm
```

Lanzamos los servicios:

```
Ejecutar como usuario root

[root@node01 ~]# systemctl start messagebus.service
[root@node01 ~]# systemctl start libvirtd.service
[root@node01 ~]# systemctl start nfs.service
```

3.3.3. Configuración de la Red

Es necesario tener Interfaz de red conectado a un "Bridge". Para ello, vamos al directorio:

```
Ejecutar como usuario root

[root@node01 ~]# cd /etc/sysconfig/network-scripts/
```

Vemos cual es nuestra interfaz de red, que debe ser del tipo `ifcfg-¡nombre!`, por ejemplo: `ifcfg-enp11s0`

Así que usamos este fichero para cargar la siguiente configuración:

```
Ejecutar como root

[root@node01 ~]# pico /etc/sysconfig/network-scripts/ifcfg-enp11s0

->Cambiar todas las lineas e introducir lo siguiente:

DEVICE=ifcfg-enp11s0
BOOTPROTO=none
NM_CONTROLLED=no
ONBOOT=yes
TYPE=Ethernet
BRIDGE=br0
```

Creamos un nuevo fichero en `/etc/sysconfig/network-scripts/` llamado `ifcfg-br0` y añadimos las siguientes lineas al fichero:

```
Ejecutar como root

pico /etc/sysconfig/network-scripts/ifcfg-br0

Copiamos los siguiente:

DEVICE=br0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=dhcp
NM_CONTROLLED=no
```

Este código se usa si estamos trabajando con *DHCP*, pero si usamos una IP fija para el servidor:

```
Ejecutar como root

pico /etc/sysconfig/network-scripts/ifcfg-br0

Copiamos los siguiente:

DEVICE=br0
TYPE=Bridge
IPADDR=<TU IP FIJA>
NETMASK=<TU MASCARA DE RED>
ONBOOT=yes
BOOTPROTO=static
NM_CONTROLLED=no
```

Después de estos cambios reiniciamos el servicio de red:

```
systemctl restart network.service
```

3.3.4. Configuración de NFS

Si estás usando un único servidor para el Front-End y los nodos de trabajo, no tienes que realizar estos pasos.

Añade la siguiente línea a la tabla de sistemas de ficheros montados:

```
Como usuario root:

pico /etc/fstab

Copiamos al final del fichero:

192.168.1.1:/var/lib/one/ /var/lib/one/ nfs soft,intr,rsize=8192,wsiz=8192,noauto
```

En lugar de 192.168.1.1, debes indicar la dirección IP del servidor donde esta instalado el Front-End

Hecho esto, ahora hay que probar montar el directorio que hemos configurado ejecutando:

```
Como usuario root:

mount /var/lib/one/
```

Si al ejecutar este comando, no responde, o muestra algún tipo de error se debe probablemente a que no se desactivo el *Firewall* y *SELinux*. Para ello vuelva a 3.1 si no lo ha hecho.

3.4. Pasos finales y verificaciones

Como se indicó en 3.2.5, es necesario que se pueda acceder a todos los nodos y desde todos los nodos sin necesidad de clave para *ssh*. Para ello tenemos que realizar las siguientes instrucciones.

Vamos al servidor principal:

```
Como usuario root:

[root@master ~]#
```

4. Despliegue y gestión de OpenNebula

4.1. Registrar nodos en OpenNebula

En los pasos anteriores hemos configurado nodos con OpenNebula tanto para Servidor como para los Nodos Trabajadores (*WorkerNode*), pero aun no lo hemos registrado en el

sistema de OpenNebula.

Registrar un nuevo nodo trabajador en OpenNebula permite que el nuevo nodo forme parte de la constelación de OpenNebula y pueda ser gestionado y utilizado por el sistema .

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# onehost create localhost -i kvm -v kvm -n dummy
```

Esto habilita el nodo '*localhost*' como nodo que gestiona OpenNebula. Si tuviésemos mas nodos, se añadirían del mismo nodo, indicando el nombre del nodo o la *IP*. Estos serian ejemplos validos:

```
[oneadmin@master ~]# onehost create node02 -i kvm -v kvm -n dummy
[oneadmin@master ~]# onehost create 192.168.1.3 -i kvm -v kvm -n dummy
```

Cada vez que usamos esta sentencia para un nodo concreto, estamos añadiendo todos los recursos de esa máquina a nuestra gestión con OpenNebula.

4.2. Añadir un Interfaz de Red Virtual

Para poder utilizar OpenNebula inicialmente es necesario realizar la siguiente operación que permitirá crear una Interfaz de Red Virtual para las máquinas virtuales que se desplegarán en esta fase de pruebas.

Esta operación también se puede realizar desde el entorno web de OpenNebula.

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# pico /tmp/mired.one

Copiar lo siguiente:

NAME = "private"

BRIDGE = br0

AR = [
    TYPE = IP4,
    IP = 192.168.0.100,
    SIZE = 3
]
```

Se guarda el archivo. Aquí lo que hemos hecho es crear un recurso correspondiente a una RED Virtual para las Máquinas Virtuales que se instale, de modo que las máquinas tendrán *IP* del tipo *192.168.0.100* con un máximo de 3 ip's posibles.

Ahora, creamos la red:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# onevnet create /tmp/mired.one
```

4.3. Usuario oneadmin

Cuando se instalo el servidor, se generó el usuario *oneadmin* y también se creo la clave para este usuario.

Esta clave la necesitamos para poder trabajar con entorno web.

Para consultar la clave hacemos lo siguiente:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# cat .one/one_auth
oneadmin:XXXXXXXXXXXXXXXXXXXX
```

De modo que la clave es la parte XXXXXXXXXXXXXXXXXXXX .

4.4. Gestión de OpenNebula a través del entorno web

Para la gestión de OpenNebula se pueden utilizar de dos modos diferentes, uno a través de una interfaz web y otro a través de la línea de comandos.

En este apartado veremos como gestionar el sistema de OpenNebula a través del entorno web. Para ello hay que acceder a través de un navegador web a la dirección dentro del servidor donde hemos instalado el Front-End:

`http://localhost:9869`

Nos pedirá la autenticación. El nombre usuario es: *oneadmin* y la clave es la que nos aparece siguiendo los pasos del apartado anterior.

Ten en cuenta que si quieres acceder al entorno web desde otra máquina, en lugar de usar *localhost*, deberás indicar la IP o el nombre de la máquina donde esté instalado el Front-End. Por ejemplo, si el servidor es **192.168.1.1**, usamos:

`http://192.168.1.1:9869`

También hay que recordar que si estás ejecutando OpenNebula dentro de una máquina Virtual con *VirtualBox*, *VMWare*, etc. hay que abrir el puerto *9869* y hacer "*Port Forwarding*", de modo que cuando se llame a la dirección web desde anfitrión sepa que el puerto *9869* se provee desde la máquina Virtual OpenNebula y por tanto se podrá acceder al entorno de gestión de OpenNebula.

En cambio si se esta ejecutando OpenNebula de manera nativa, simplemente con hacer *IP Forwarding* y aceptar conexiones para el puerto *9869*.

4.4.1. Interfaz de Web, panel de control

La primera pantalla que obtenemos es el panel de control completo de OpenNebula:

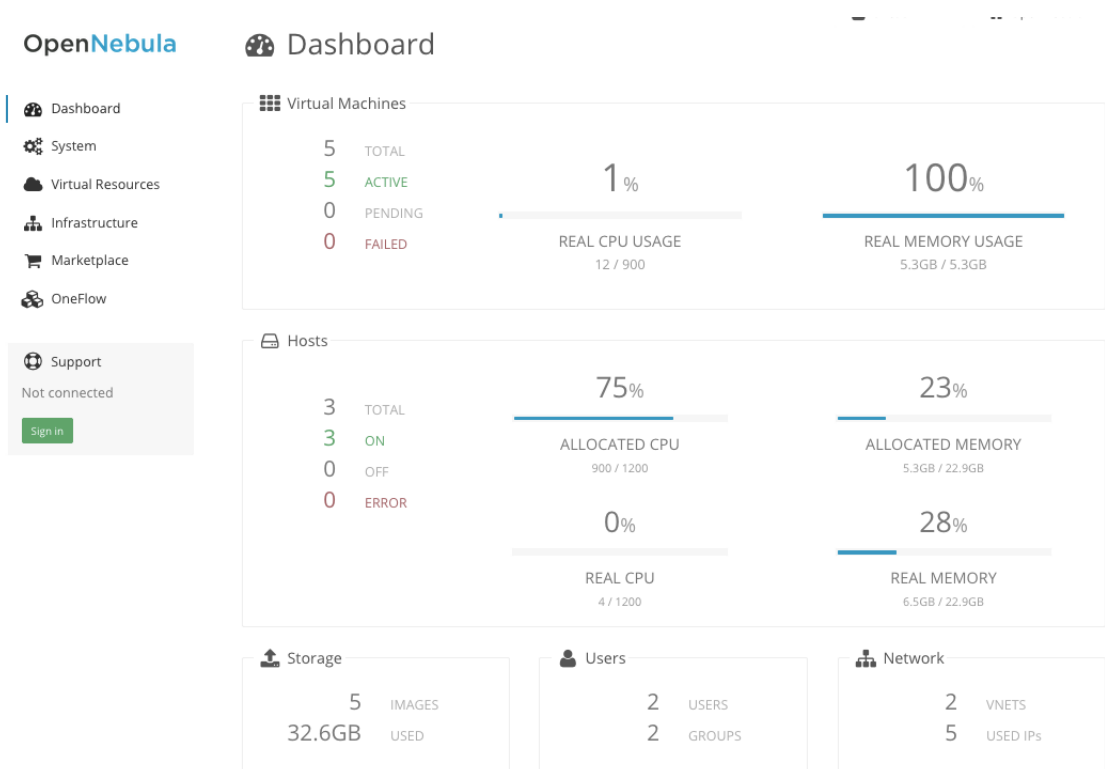


Figura 4: DashBoard de OpenNebula

En la parte central de la imagen se observa el resumen del funcionamiento de todas las máquinas virtuales que están registradas en OpenNebula.

Como se observa hay varias opciones para gestionar todos los aspectos que hemos visto en el sistema y en la arquitectura de OpenNebula. Las operaciones que permite la gestión del entorno web de OpenNebula es:

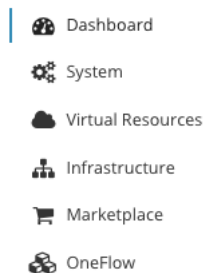


Figura 5: Opciones de gestión de OpenNebula

- **Dashboard.** Pantalla con los detalles de todo el sistema
- **System.** Permite gestionar los usuarios, grupos, etc para el acceso a los servicios de OpenNebula
- **VirtualResources.** Permite la gestión de Máquinas Virtuales, Plantillas de creación de máquinas virtuales , Imágenes de Máquinas Virtuales y Ficheros.
- **Infraestructura.** Permite la gestión de *Clusters*, *Hosts*, *Datastores*, Redes Virtuales, zonas, ...

- **MarketPlace. Importar una imagen.** Repositorio o almacén en línea de imágenes de Sistemas Operativos, listas para poder ser utilizadas dentro de OpenNebula.

4.5. MaketPlace

La opción de *MarketPlace* ofrece un conjunto de imágenes preparadas listas para ser usadas con OpenNebula, y directamente integrables en el sistema. De modo que si por ejemplo queremos probar *CentOS7*, *Hadoop*, *UbuntuServer*, ... simplemente con seleccionar la imagen y automáticamente se integra dentro de OpenNebula.

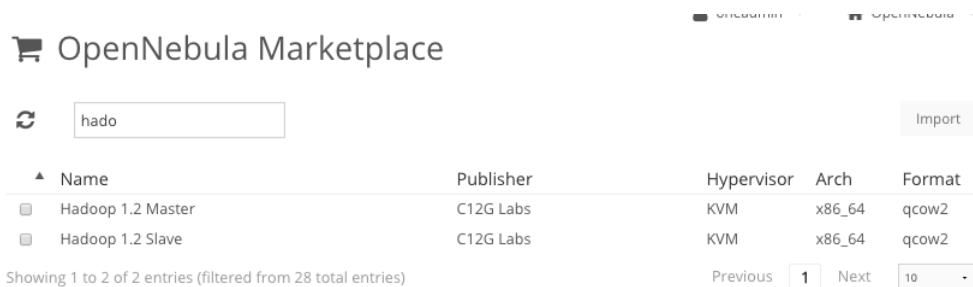


Figura 6: Búsqueda de una imagen de Hadoop

Ahora escribimos en la caja de búsqueda "*Debian*".

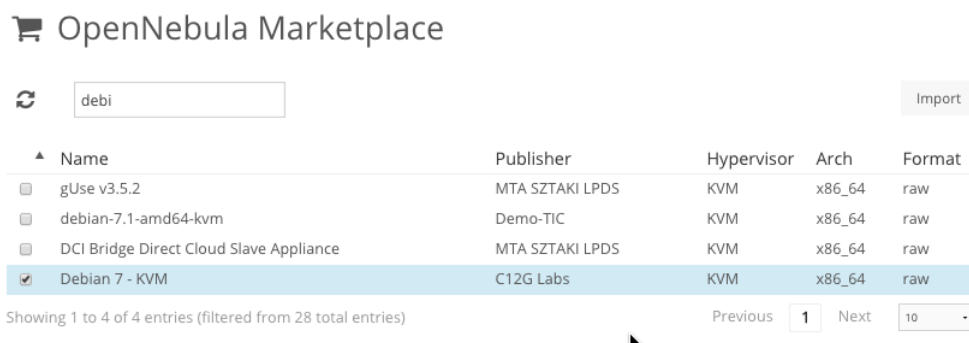


Figura 7: Selección e importación de la imagen de Debian 7

Hacemos '*clic*' en *Import* y pide la confirmación sobre la importación, volvemos a hacer '*clic*' en *Import*.

Una vez que termine el proceso vamos a la opción del menú de la izquierda **Virtual Resources** —> **Images**, donde veremos la nueva imagen ya integrada en el sistema:

ID	Owner	Group	Name	Datastore	Type	Status	#VMS
10	oneadmin	oneadmin	Debian7_test	default	OS	LOCKED	0
9	oneadmin	oneadmin	CentOS-6.5-one-4.8	default	OS	USED	3
8	oneadmin	oneadmin	Hadoop 1.2 Slave_file0	default	OS	READY	0
6	oneadmin	oneadmin	Hadoop 1.2 Master_file0	default	OS	READY	0
5	oneadmin	oneadmin	Debian-7	default	OS	USED	1
0	oneadmin	oneadmin	CentOS-7-one-4.8	default	OS	USED	1

Showing 1 to 6 of 6 entries

6 TOTAL 42.6GB USED

Figura 8: Imagen importada lista para utilizarla

4.6. Despliegue de la máquina virtual

Una vez importada la imagen de **Debian** y también que el estado de la imagen es **READY**, podemos disponer de la imagen para poder despegarla.

Cada una de las imágenes puede llevar asociada una plantilla de creación de Máquina Virtual. Esto permite crear máquinas virtuales preconfiguradas con unos parámetros por defecto de modo que todas las máquinas virtuales que se desplieguen con esa plantilla tendrán atributos comunes.

Plantillas

Las plantillas permiten desplegar Máquinas virtuales con atributos ya predefinidos con unas determinadas características. Por ejemplo si queremos desplegar muchas máquinas virtuales correspondientes a nodos sencillos que van a tener las siguientes características:

- 2 *CPU*s
- 1GB de *RAM*
- Red del tipo 192.168.1.100 con un rango de 10 direcciones *IP*.

Se puede crear una plantilla con estos atributos, de modo que al desplegar las máquinas virtuales, se usa esta plantilla que define este grupo de características.

En nuestro caso usaremos la plantilla por defecto que se genero al importar la imagen de *Debian*:

ID	Owner	Group	Name	Registration time
13	oneadmin	oneadmin	Debian 7 - KVM_test	19:11:57 23/02/2015
12	oneadmin	oneadmin	Centos65Slave_02	11:42:57 17/02/2015
11	oneadmin	oneadmin	Centos65Slave	11:41:32 17/02/2015
10	oneadmin	oneadmin	CentOS 6.5 - KVM	09:17:14 17/02/2015
9	oneadmin	oneadmin	CentOS7-SlaveHadoop	11:38:00 16/02/2015
8	oneadmin	oneadmin	CentOS7-MasterHadoop	11:36:34 16/02/2015
4	oneadmin	oneadmin	Debian 7 - KVM	18:33:19 12/02/2015
0	oneadmin	oneadmin	CentOS-7	13:58:26 11/02/2015

Showing 1 to 8 of 8 entries

Figura 9: Plantillas instaladas. Seleccionada la última plantilla que se ha instalado al importar *Debian*

En esta plantilla hay que realizar una modificación para que pueda tomar la red virtual que se habia creado. Para ello seleccionamos la plantilla y se hace 'clic' en 'Update'. Vamos a la opción Networks y seleccionamos la red virtual que se creó:

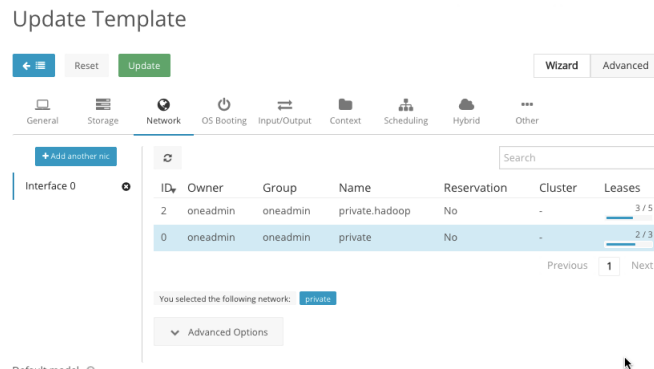


Figura 10: Red seleccionada para la plantilla

Una vez seleccionada, se hace 'clic' de nuevo en 'Update', para guardar lo valores de la plantilla con los cambios de la red.

4.6.1. Creación de máquina virtual

Para ello vamos al menú '*Virtual Resources*' – > '*Virtual Machines*'.

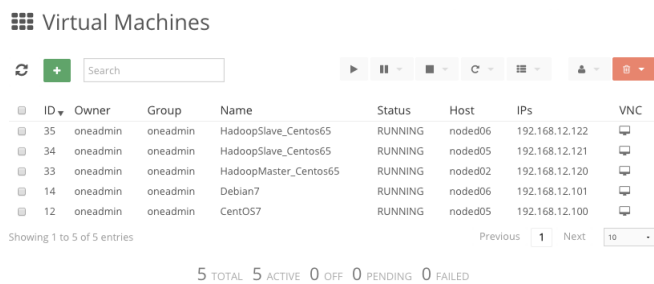


Figura 11: Máquinas virtuales instaladas

En la figura anterior podemos ver todas las máquinas virtuales desplegadas en el nuestro sistema OpenNebula

Ahora añadimos una nueva máquina virtual. Para ello hacemos 'clic' en el símbolo:



Figura 12: Botón para añadir una o varias máquinas virtuales

Donde se despliega un menú para la creación de la máquina virtual.

Los datos necesarios son:

- Nombre de la Máquina Virtual
- Número de Instancias
- Plantilla (*Template*) para generar la Máquina Virtual

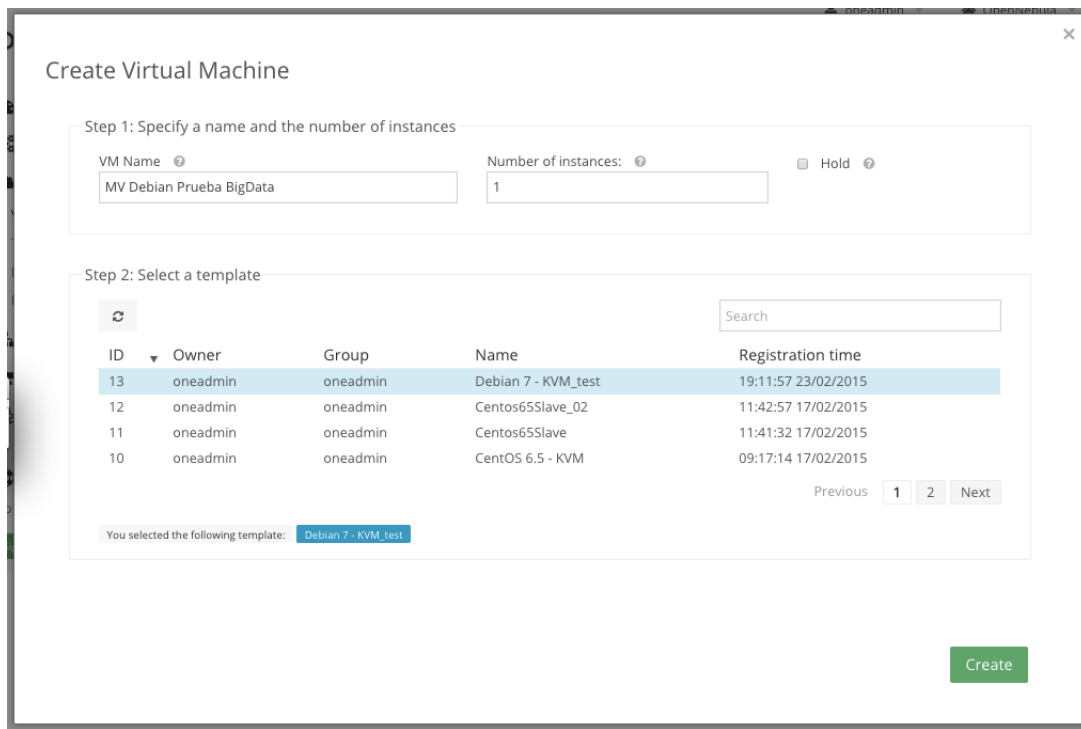


Figura 13: Creación de la Máquina Virtual

Una vez hecho esto la máquina virtual comienza a iniciarse, y se pueden ver los estados de la máquina virtual.

The screenshot shows the VM status interface. At the top, there are control buttons (refresh, add, search, play, pause, stop, restart, etc.). Below is a table with the following data:

ID	Owner	Group	Name	Status	Host	IPs	VNC
37	oneadmin	oneadmin	MV Debian BigData	RUNNING	noded05	192.168.12.102	[VNC icon]

Figura 14: Estado de la máquina virtual. *RUNNING* e *IP* asignada

Una vez iniciada la máquina virtual, el sistema le asigna la *IP* según el *Template* seleccionado. Como se observa en la imagen anterior la *IP* que se le ha asignado es *192.168.12.102*.

Con esto la máquina virtual ya está funcionando y lista para poder trabajar con ella, pero ahora veremos como acceder a la máquina virtual en el siguiente apartado.

4.6.2. Acceso a la máquina virtual

Hecho el paso anterior, ya tenemos los recursos de la máquina virtual disponibles para ser usados.

Para el acceso a esta nueva máquina virtual que hemos creado volvemos a un terminal dentro de la máquina servidor Front-End de OpenNebula:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# ssh root@192.168.12.102
```

En nuestro caso la dirección IP que le ha asignado OpenNebula es 192.168.12.102.

Con esto nos conectamos a la nueva máquina virtual que está ejecutando la imagen de *CentOS*:

```
Warning: Permanently added '192.168.12.102' (ECDSA) to the list of known hosts.
Linux debian 3.2.0-4-amd64 #1 SMP Debian 3.2.60-1+deb7u3 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~#
```

Por tanto ya tenemos disponible esta máquina virtual ejecutándose y disponible para funcionar perfectamente.

5. Despliegue y gestión de OpenNebula utilizando la línea de comandos

Toda la sección anterior explica como se puede desplegar OpenNebula utilizando el entorno Web que proporciona SunStone y que permite gestionar todo el sistema desde un entorno muy sencillo e intuitivo y de alto nivel.

Existe la posibilidad de realizar las mismas operaciones que se realizan dentro del entorno web pero desde un interfaz para administradores usando la línea de comandos del *shell* del sistema donde se instala el Front-End.

Vamos a realizar los mismos pasos que se usan en la sección anterior pero utilizando los comandos de administración.

5.1. Registro de nodos

Igual que en la sección de despliegue web se usa el comando que permite añadir nuevos nodos al sistema:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# onehost create localhost -i kvm -v kvm -n dummy
```

Esto habilita el nodo 'localhost' como nodo que gestiona OpenNebula. Si tuviésemos mas nodos, se añadirían del mismo nodo, indicando el nombre del nodo o la IP. Estos serian ejemplos validos:

```
[oneadmin@master ~]# onehost create node02 -i kvm -v kvm -n dummy
[oneadmin@master ~]# onehost create 192.168.1.3 -i kvm -v kvm -n dummy
```

Cada vez que usamos esta sentencia para un nodo concreto, estamos añadiendo todos los recursos de esa máquina a nuestra gestión con OpenNebula.

5.2. Añadir un Interfaz de Red Virtual

Para poder utilizar OpenNebula inicialmente es necesario realizar la siguiente operación que permitirá crear una Interfaz de Red Virtual para las máquinas virtuales que se desplegarán en esta fase de pruebas.

Esta operación también se puede realizar desde el entorno web de OpenNebula.

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# pico /tmp/mired.one

Copiar lo siguiente:
```

```
NAME = "private"

BRIDGE = br0

AR = [
    TYPE = IP4,
    IP = 192.168.0.100,
    SIZE = 3
]
```

Se guarda el archivo. Aquí lo que hemos hecho es crear un recurso correspondiente a una RED Virtual para las Máquinas Virtuales que se instale, de modo que las máquinas tendrán *IP* del tipo *192.168.0.100* con un máximo de 3 ip's posibles.

Ahora, creamos la red:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# onevnet create /tmp/mired.one
```

5.3. Usuario oneadmin

Cuando se instalo el servidor, se generó el usuario *oneadmin* y también se creo la clave para este usuario.

Esta clave la necesitamos para poder trabajar con entorno web.

Para consultar la clave hacemos lo siguiente:

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# cat .one/one_auth
oneadmin:XXXXXXXXXXXXXXXXXXXX
```

De modo que la clave es la parte XXXXXXXXXXXXXXXXXXXXXXX .

5.4. Descargar una imagen del MarketPlace

Para utilizar una imagen usamos el comando: *oneimage create*

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# oneimage create --name "CentOS-7-one-4.8" \
    --path http://marketplace.ci2g.com/appliance/53e7bf928fb81d6a69000002/download \
    --driver qcow2 \
    -d default
```

5.5. Creación de plantilla para la imagen descargada

Para crear una plantilla específica con los atributos de memoria, *cpu*, red, etc. se usa el comando:

onetemplate create

```
Como usuario oneadmin:

[root@master ~]# su - oneadmin

[oneadmin@master ~]# onetemplate create --name "CentOS-7" \
    --cpu 1 --vcpu 1 --memory 512 --arch x86_64 \
    --disk "CentOS-7-one-4.8" \
    --nic "private" \
    --vnc --ssh --net_context
```

5.6. Ejecución de la Máquina Virtual

Para ejecutar la máquina virtual usamos el comando: *onetemplate instantiate*

```
Como usuario oneadmin:  
[root@master ~]# su - oneadmin  
[oneadmin@master ~]# onetemplate instantiate "CentOS-7"
```

5.7. Consulta de la lista de máquinas virtuales

```
Como usuario oneadmin:  
[root@master ~]# su - oneadmin  
[oneadmin@master ~]# onevm list
```

5.8. Lista de comando útiles

Consulta la lista de plantillas:

```
[oneadmin@master ~]# onetemplate list
```

Consulta la lista de redes virtuales disponibles:

```
[oneadmin@master ~]# onevnet list
```

Consulta la lista de imágenes instaladas del *MarketPlace*

```
[oneadmin@master ~]# oneimage list
```

Otros comandos disponibles:

```
oneacct: Muestra la informacion de cuentas de usuario.  
oneacl: Controla la lista de usuarios  
onecluster: Gestiona los cluster.  
onedatastore: Gestiona los datastores  
onedb: Control de la herramienta de migracion de la BBDD  
onegroup: Gestiona los grupos  
onehost: Gestiona los Hosts  
oneimage: Gestiona las imagenes  
onetemplate: Gestiona los templates  
oneuser: Gestiona los usuarios  
onevdc: Gestiona los Data Centers Virtuales  
onevm: Gestiona las maquinas virtuales  
onevnet: Gestiona las redes  
onezone: Gestiona las zonas
```

6. Bibliografia

- OpenNebula Documentation. <http://opennebula.org/documentation/>
- OpenNebula Command Line Interface. <http://docs.opennebula.org/4.8/user/references/cli.htm>
- OpenNebula 4.4 Intallation Guide. <http://archives.opennebula.org/documentation:rel4.4:ignc>
- OpenNebula Administration. <http://docs.opennebula.org/4.10/administration/>
- OpenNebula Features. <http://opennebula.org/about/key-features/>