

Contenido de la guía

INFORMACIÓN ACADÉMICA DE LA GUÍA..... 2

    OBJETIVOS: ..... 2

    SOFTWARE A UTILIZAR: ..... 2

    CONCEPTOS FUNDAMENTALES ..... 2

I. INSTALACIÓN DE HIPERVISOR KVM..... 3

    1 – CONFIGURACIÓN DE RED ..... 3

        Paso 0 – Pasos previos ..... 3

        Paso 1 – Desactivar SELINUX ..... 3

        Paso 2 – Instalar hipervisor KVM ..... 3

        Paso 3 – Configuración de la red ..... 5

    2 – INSTALACIÓN DE OPENNEBULA ..... 8

        Paso 1 – Creación del repositorio ..... 8

        Paso 2 – Instalación del software para nodo de OpenNebula ..... 8

Material elaborado por Víctor Cuchillac (padre).

Objetivos:

- Instalar el hipervisor KVM en Centos.
- Crear una máquina virtual en KVM

Software a utilizar:

- Kitty (recomendado) Puty, si no descarga kitty
- VirtualBox en su versión más reciente (de preferencia)
- Máquina virtual hipervisor1 con archivo de configuración e imagen del disco duro de Centos 6.3 de 64 bits instalado

Conceptos Fundamentales

Una máquina virtual dentro del sistema de OpenNebula consiste en:

- Un espacio en términos de memoria y CPU
- Un grupo de tarjetas de red conectados a una o más redes virtuales
- Un conjunto de imágenes de disco
- Un archivo de estado (opcional) o archivo de recuperación del estado, el cual contiene la imagen de la memoria de una máquina virtual corriendo además de información específica del hipervisor.

Los elementos anteriores, además de algunos atributos VM adicionales como el núcleo del sistema operativo y la información de contexto para ser utilizado dentro de la VM, se especifican en un archivo de plantilla.

Tipos de Discos. En cada disco es definido un atributo DISK. Una máquina virtual puede utilizar tres tipos de disco:

- **Imagen persistente (persistent image):** Los cambios en la imagen del disco continuarán después que de la VM se haya apagado.
- **Imagen no-persistente (non-persistent image):** Las imágenes son clonadas, los cambios en la imagen se perderán.
- **Discos volátiles:** se crean sobre la marcha en el host de destino.

# I. Instalación de hipervisor1 KVM

## 1 – Configuración de red

### Paso 0 – Pasos previos

0.1 Verifique que exista comunicación a Internet

0.2 Ingrese como usuario root

### Paso 1 – Desactivar SELINUX

#### 1.1 Sustituir la opción enforcing por disabled

```
[root@hipervisor1 ~]# mcedit /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

#### 1.2 Reiniciar el equipo

```
[root@hipervisor1 ~]# reboot
```

### Paso 2 – Instalar hipervisor KVM

#### 2.1 Importar las llaves GPG para los paquetes a instalar

```
[root@hipervisor1 ~]# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
```

#### 2.2 Instalar el hipervisor KVM y las herramientas de configuración

```
[root@hipervisor1 ~]# yum install kvm libvirt python-virtinst qemu-kvm
```

```
Loaded plugins: fastestmirror, refresh-packagekit, security
Determining fastest mirrors
 * base: mirror.unl.edu
 * extras: mirrors.mit.edu
 * updates: mirrors.xmission.com
base                               | 3.7 kB    00:00
base/primary_db                    | 4.6 MB    00:13
extras                              | 3.4 kB    00:00
extras/primary_db                  | 26 kB     00:00
opennebula                          | 2.9 kB    00:00
opennebula/primary_db              | 20 kB     00:00
updates                             | 3.4 kB    00:00
updates/primary_db                 | 1.3 MB    00:03
Setting up Install Process
Resolving Dependencies
--> Running transaction check
. . .
. . .
Dependencies Resolved
```

```

=====
Package                Arch                Version                Repository            Size
=====
Installing:
libjpeg-turbo          x86_64              1.2.1-3.el6_5         base                  174 k
  replacing libjpeg.x86_64 6b-46.el6
libvirt                 x86_64              0.10.2-54.el6         base                  2.4 M
python-virtinst        noarch              0.600.0-29.el6        base                  538 k
qemu-kvm                x86_64              2:0.12.1.2-2.479.el6 base                  1.6 M
Installing for dependencies:
augeas-libs            x86_64              1.0.0-10.el6          base                  314 k
celt051                 x86_64              0.5.1.3-0.el6         base                   50 k
cyrus-sasl-md5         x86_64              2.1.23-15.el6_6.2    base                   47 k
eatables               x86_64              2.0.9-6.el6           base                   95 k
glusterfs              x86_64              3.6.0.54-1.el6       base                  1.3 M
glusterfs-api          x86_64              3.6.0.54-1.el6       base                   62 k
glusterfs-libs         x86_64              3.6.0.54-1.el6       base                   272 k
gnutls-utils           x86_64              2.8.5-18.el6          base                   101 k
gppe-roms-qemu         noarch              0.9.7-6.14.el6        base                   221 k
libcgroup              x86_64              0.40.rc1-16.el6       base                   129 k
libvirt-client         x86_64              0.10.2-54.el6         base                   4.1 M
libvirt-python         x86_64              0.10.2-54.el6         base                   500 k
lzo                    x86_64              2.03-3.1.el6_5.1     base                   55 k
lzop                   x86_64              1.02-0.9.rc1.el6      base                   50 k
nc                     x86_64              1.84-24.el6           base                   57 k
netcf-libs             x86_64              0.2.4-3.el6           base                   64 k
. . .
. . .
polkit                 x86_64              0.96-11.el6           base                   162 k
zlib                   x86_64              1.2.3-29.el6          base                   73 k
=====

```

Transaction Summary

```

=====
Install      31 Package(s)
Upgrade      8 Package(s)
=====

```

Total download size: 16 M

Is this ok [y/N]: **y** ← ←

Downloading Packages:

```

(1/39): augeas-libs-1.0.0-10.el6.x86_64.rpm | 314 kB    00:01
(2/39): celt051-0.5.1.3-0.el6.x86_64.rpm   | 50 kB     00:00
(3/39): cyrus-sasl-2.1.23-15.el6_6.2.x86_64.rpm | 78 kB     00:00
. . .

```

```

Verifying : polkit-0.96-2.el6_0.1.x86_64 46/48
Verifying : cyrus-sasl-lib-2.1.23-13.el6.x86_64 47/48
Verifying : gnutls-2.8.5-4.el6_2.2.x86_64 48/48

```

Installed:

```

libjpeg-turbo.x86_64 0:1.2.1-3.el6_5      libvirt.x86_64 0:0.10.2-54.el6
python-virtinst.noarch 0:0.600.0-29.el6      qemu-kvm.x86_64 2:0.12.1.2-2.479.el6

```

Dependency Installed:

```

augeas-libs.x86_64 0:1.0.0-10.el6      celt051.x86_64 0:0.5.1.3-0.el6
cyrus-sasl-md5.x86_64 0:2.1.23-15.el6_6.2  eatables.x86_64 0:2.0.9-6.el6
glusterfs.x86_64 0:3.6.0.54-1.el6      glusterfs-api.x86_64 0:3.6.0.54-1.el6
. . .
. . .
seabios.x86_64 0:0.6.1.2-30.el6        sgabios-bin.noarch 0:0-0.3.20110621svn.el6
snappy.x86_64 0:1.1.0-1.el6            spice-server.x86_64 0:0.12.4-12.el6
usbredir.x86_64 0:0.5.1-2.el6          vgabios.noarch 0:0.6b-3.7.el6
yajl.x86_64 0:1.0.7-3.el6

```

Dependency Updated:

```

cyrus-sasl.x86_64 0:2.1.23-15.el6_6.2  cyrus-sasl-gssapi.x86_64 0:2.1.23-15.el6_6.2
cyrus-sasl-lib.x86_64 0:2.1.23-15.el6_6.2  cyrus-sasl-plain.x86_64 0:2.1.23-15.el6_6.2
gnutls.x86_64 0:2.8.5-18.el6            openssl.x86_64 0:1.0.1e-42.el6
polkit.x86_64 0:0.96-11.el6             zlib.x86_64 0:1.2.3-29.el6

```

Replaced:

```

libjpeg.x86_64 0:6b-46.el6

```

**Complete!**

## 2.3 Verificar que se inicia el hipervisor KVM

```
[root@hipervisor1 ~]# /etc/init.d/libvirtd start
```

```
Iniciando demonio libvirtd: [ OK ]
```

```
[root@hipervisor1 ~]# service messagebus start
```

```
Iniciando bus de mensajes del sistema:
```

## 2.4 Comprobar el funcionamiento del hipervisor KVM

```
[root@hipervisor1 ~]# virsh -c qemu:///system list
```

```
Id      Nombre                               Estado
-----
```

Nota si todo está bien no deben aparecer errores.

## Paso 3 – Configuración de la red

### 3.1 Instalar las herramientas para bridge

```
[root@hipervisor1 ~]# yum install bridge-utils
```

```
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: mirror.unl.edu
 * extras: mirrors.mit.edu
 * updates: mirrors.xmission.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package bridge-utils.x86_64 0:1.2-9.el6 will be updated
---> Package bridge-utils.x86_64 0:1.2-10.el6 will be an update
--> Finished Dependency Resolution
```

```
. . . .
```

```
Running Transaction
  Updating   : bridge-utils-1.2-10.el6.x86_64 1/2
  Cleanup   : bridge-utils-1.2-9.el6.x86_64 2/2
  Verifying  : bridge-utils-1.2-10.el6.x86_64 1/2
  Verifying  : bridge-utils-1.2-9.el6.x86_64 2/2
```

```
Updated:
  bridge-utils.x86_64 0:1.2-10.el6
```

```
Complete!
```

### 3.2 Crear una copia del archivo de configuración de la tarjeta de red

Digitar en una sola línea:

```
[root@hipervisor1 ~]# cp /etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth0.original
```

### 3.3 Verificar que se haya creado una copia del archivo

Digitar en una sola línea:

```
[root@hipervisor1 ~]# ls /etc/sysconfig/network-scripts/ifcfg-eth0*  
  
/etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0.original
```

### 3.4 Editar las opciones de la tarjeta de red eth0

```
[root@hipervisor1 ~]# mcedit /etc/sysconfig/network-scripts/ifcfg-eth0
```

☞**Nota:** Cada equipo debe tener su propia dirección MAC, si aparece eth1 en lugar de eth0, porque ha cambiado la dirección MAC después de haber iniciado el sistema operativo, puede borrar el archivo: y reiniciar el equipo

```
DEVICE=eth0  
BOOTPROTO=none  
NM_CONTROLLED=no  
ONBOOT=yes  
TYPE=Ethernet  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=yes  
IPV6INIT=no  
NAME="System eth0"  
USERCTL=no  
BRIDGE=br 0
```

### 3.5 Editar las opciones de la tarjeta br0

☞**Nota:** Tenga en cuenta que para el laboratorio se utilizará DHCP, en un escenario real sería estática

```
[root@hipervisor1 ~]# mcedit /etc/sysconfig/network-scripts/ifcfg-br0
```

```
DEVICE=br0  
BOOTPROTO=dhcp  
NM_CONTROLLED=no  
ONBOOT=yes  
TYPE=Bridge  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=yes  
IPV6INIT=no  
NAME="System br 0"  
USERCTL=no
```

### 3.6 Reinicie el servicio de red

```
[root@hipervisor1 ~]# service network restart

Interrupción de la interfaz br0: [ OK ]
Interrupción de la interfaz eth0: [ OK ]
Interrupción de la interfaz de loopback: [ OK ]
Activación de la interfaz de loopback: [ OK ]
Activando interfaz eth0: [ OK ]
Activando interfaz br0: [ OK ]
Determinando la información IP para br0... hecho. [ OK ]
```

### 3.7 Detener servicio gráfico de configuración de la tarjeta de red

Esto se hará para evitar que el NetworkManager (herramienta gráfica de configuración de las tarjetas de red), pueda interferir en algún momento con la configuración de las tarjetas. Es opcional, pero si se tiene el escritorio ejecutándose de forma predeterminada se inicia el NetworkManager

☞ **Nota:** Con la instrucción `chkconfig` definimos si un servicio inicia o no durante el arranque del sistema.

```
[root@hipervisor1 ~]# service NetworkManager stop

Parando el demonio NetworkManager: [ OK ]

[root@hipervisor1 ~]# chkconfig NetworkManager off

[root@hipervisor1 ~]# chkconfig network on

[root@hipervisor1 ~]# service network start

Activación de la interfaz de loopback: [ OK ]
Activando interfaz eth0: device eth0 is already a member of a bridge; can't enslave
it to bridge br0. [ OK ]
Activando interfaz br0:
```

### 3.7 Verifique que se hayan creado los objetos de red

```
[root@hipervisor1 ~]# ifconfig

br0      Link encap:Ethernet  HWaddr 08:00:AA:00:00:11
         inet addr:192.168.20.107  Bcast:IP del equipo  Mask:255.255.255.0
         inet6 addr: fe80::a00:aaff:fe00:11/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:775 errors:0 dropped:0 overruns:0 frame:0
         TX packets:399 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:60554 (59.1 KiB)  TX bytes:48322 (47.1 KiB)

eth0     Link encap:Ethernet  HWaddr 08:00:AA:00:00:11
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:1500 errors:0 dropped:0 overruns:0 frame:0
         TX packets:679 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:245585 (239.8 KiB)  TX bytes:78214 (76.3 KiB)
```

```

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:87 errors:0 dropped:0 overruns:0 frame:0
            TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:6614 (6.4 KiB)  TX bytes:6614 (6.4 KiB)

virbr0     Link encap:Ethernet  HWaddr 52:54:00:DF:F6:DA
            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:11 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 b)  TX bytes:2549 (2.4 KiB)

```

## 2 – Instalación de OpenNebula

☞ **Nota:** En esta guía se ha instalado la versión 4.12, pero se puede instalar la versión 4.14 o superior.

### Paso 1 – Creación del repositorio

#### 1.1 Crear un nuevo repositorio de paquetes

```
[root@hipervisor1 ~]# cat /etc/yum.repos.d/opennebula.repo
```

#### 1.2 Definir el siguiente contenido

```

[opennebula]
name=opennebula
baseurl=http://downloads.opennebula.org/repo/4.12/CentOS/6/x86_64
enabled=1
gpgcheck=0

```

### Paso 2 – Instalación del software para nodo de OpenNebula

```
[root@hipervisor1 ~]# yum install opennebula-node-kvm
```

```

Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: mirror.unl.edu
 * extras: mirrors.mit.edu
 * updates: mirrors.xmission.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package opennebula-node-kvm.x86_64 0:4.12.1-1 will be installed
. . .
. . .
--> Running transaction check
---> Package compat-readline5.x86_64 0:5.2-17.1.el6 will be installed
--> Finished Dependency Resolution

```



Dependencies Resolved

```
=====
Package                Arch      Version      Repository    Size
=====
Installing:
  opennebula-node-kvm  x86_64    4.12.1-1     opennebula    8.1 k
Installing for dependencies:
  compat-readline5     x86_64    5.2-17.1.el6 base           130 k
  ipset                 x86_64    6.11-4.el6   base           63 k
  libmnl                x86_64    1.0.2-3.el6  base           21 k
  opennebula-common    x86_64    4.12.1-1     opennebula    14 k
  ruby                  x86_64    1.8.7.374-4.el6_6 base           538 k
  ruby-libs             x86_64    1.8.7.374-4.el6_6 base           1.7 M
=====
```

Transaction Summary

```
=====
Install      7 Package(s)
```

Total download size: 2.4 M

Installed size: 8.1 M

Is this ok [y/N]: y

Downloading Packages:

(1/7): compat-readline5-5.2-17.1.el6.x86_64.rpm	130 kB	00:00
(2/7): ipset-6.11-4.el6.x86_64.rpm	63 kB	00:00
(3/7): libmnl-1.0.2-3.el6.x86_64.rpm	21 kB	00:00
(4/7): opennebula-common-4.12.1-1.x86_64.rpm	14 kB	00:00
(5/7): opennebula-node-kvm-4.12.1-1.x86_64.rpm	8.1 kB	00:00
(6/7): ruby-1.8.7.374-4.el6_6.x86_64.rpm	538 kB	00:02
(7/7): ruby-libs-1.8.7.374-4.el6_6.x86_64.rpm	1.7 MB	00:04

Total

253 kB/s

| 2.4 MB 00:09

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing	: libmnl-1.0.2-3.el6.x86_64	1/7
Installing	: ipset-6.11-4.el6.x86_64	2/7
Installing	: opennebula-common-4.12.1-1.x86_64	3/7
Installing	: compat-readline5-5.2-17.1.el6.x86_64	4/7
Installing	: ruby-libs-1.8.7.374-4.el6_6.x86_64	5/7
Installing	: ruby-1.8.7.374-4.el6_6.x86_64	6/7
Installing	: opennebula-node-kvm-4.12.1-1.x86_64	7/7
Verifying	: compat-readline5-5.2-17.1.el6.x86_64	1/7
Verifying	: opennebula-node-kvm-4.12.1-1.x86_64	2/7
Verifying	: opennebula-common-4.12.1-1.x86_64	3/7
Verifying	: ruby-libs-1.8.7.374-4.el6_6.x86_64	4/7
Verifying	: ipset-6.11-4.el6.x86_64	5/7
Verifying	: ruby-1.8.7.374-4.el6_6.x86_64	6/7
Verifying	: libmnl-1.0.2-3.el6.x86_64	7/7

Installed:

opennebula-node-kvm.x86\_64 0:4.12.1-1

Dependency Installed:

compat-readline5.x86\_64 0:5.2-17.1.el6 ipset.x86\_64 0:6.11-4.el6 libmnl.x86\_64  
0:1.0.2-3.el6  
opennebula-common.x86\_64 0:4.12.1-1 ruby.x86\_64 0:1.8.7.374-4.el6\_6  
ruby-libs.x86\_64 0:1.8.7.374-4.el6\_6

**Complete!**

### Paso 1 – Exportar el recurso NAS del frontend1

#### 1.1 Ingresar al frontend1

Es de recordar que es la máquina con OpenNebula 4.X

#### 1.2 Exportar el directorio /var/lib/one

Agregue la siguiente línea al archivo exports (/var/lib/one/ \*(rw, sync, no\_subtree\_check, root\_squash), puede utilizarse nano, mcedit o vi

```
[root@frontend1 ~]# mcedit /etc/exports

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

#### 1.3 Reiniciar los servicios de exportación NFS

```
[root@frontend1 ~]# service rpcbind restart

Stopping rpcbind: [ OK ]
Starting rpcbind: [ OK ]

[root@frontend1 ~]# service nfs restart

Shutting down NFS daemon: [ FAILED ]
Shutting down NFS mountd: [ FAILED ]
Starting NFS services: [ OK ]
Starting NFS mountd: [ OK ]
Stopping RPC idmapd: [ OK ]
Starting RPC idmapd: [ OK ]
Starting NFS daemon: [ OK ]
```

#### 1.4 Verificar arranque de servicios NFS

Debido a que se necesita que los dos servicios (nfs y rpcbind) se ejecuten en el runlevel 3 digite los siguientes comandos:

```
[root@frontend1 ~]# chkconfig |grep nfs
nfs          0:off  1:off  2:off  3:off  4:off  5:off  6:off
nfslock      0:off  1:off  2:off  3:on   4:on   5:on   6:off

[root@frontend1 ~]# chkconfig |grep rpcbind
rpcbind      0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

#### 1.5 Definir carga de NFS en el arranque

Si estuviera apagado el servicio NFS defina que se iniciará al inicio

```
[root@frontend1 ~]# chkconfig nfs on

[root@frontend1 ~]# chkconfig |grep nfs
nfs          0:off  1:off  2:on   3:on   4:on   5:on   6:off
nfslock      0:off  1:off  2:off  3:on   4:on   5:on   6:off
```

## 1.6 Verificar que se ha exportado

```
[root@frontend1 ~]# showmount -e
```

```
Export list for frontend1:
```

```
/var/lib/one/ *
```

```
[root@frontend1 ~]# exportfs -v
```

```
/var/lib/one/ <world>(rw,wdelay,no_root_squash,no_subtree_check)
```

## 1.7 Si se desea reiniciar los recursos que se exportan, digitar los siguientes comandos

Esto es útil si se ha modificado el archivo /etc/exports

```
[root@frontend1 ~]# exportfs -a
```

```
[root@frontend1 ~]# exportfs -r
```

## Paso 2 – Importar el recurso del frontend1 en el hipervisor1

### 2.1 Ingresar al hipervisor1

Es otra máquina virtual con el hipervisor KVM instalado, es la máquina 192.168.50+Y.4

### 2.2 Crear el punto de montaje

Se utilizara /var/lib/one, si ya estuviera creado omite este paso.

```
[root@hipervisor1 ~]# mkdir /var/lib/one
```

### 2.3 Asignar permisos al usuario oneadmin, creado previamente

```
[root@hipervisor1 ~]# chown oneadmin:oneadmin /var/lib/one -R
```

```
[root@hipervisor1 ~]# chmod -R 755 /var/lib/one
```

### 2.4 Verificar que se ha creado el directorio

```
[root@hipervisor1 ~]# ls -l /var/lib/
```

```
...
```

```
drwxr-xr-x. 5 root root 4096 sep 21 2012 nfs
```

```
drwxr-xr-x 2 oneadmin oneadmin 4096 dic 8 11:22 one
```

```
...
```

### 2.5 Agregar la línea de conexión para montar el directorio al archivo /etc/fstab

Nota: puede utilizarse el nombre de host frontend1 o la dirección IPv4

```
[root@hipervisor1 ~]# mcedit /etc/fstab
```

Agregar al final del archivo la siguiente línea (es una sola, debe hacerse ping al host), puede utilizar la IP

```
frontend1:/var/lib/one/ /var/lib/one nfs
```

```
soft,intr,rsize=8192,wsiz=8192,noauto 0 0
```

## 2.6 Verificar que se haya editado correctamente

```
[root@hipervisor1 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Jul 22 18:55:49 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/vg_one1-lv_root / ext4 defaults 1 1
UUID=4f1fa095-8827-4514-ac59-1dfc4fa47132 /boot ext4 defaults
1 2
/dev/mapper/vg_one1-lv_swap swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0

#/dev/sdb1 /mnt/iscsi_lun1 ext4 defaults,_netdev 0 0

frontend1:/var/lib/one/ /var/lib/one nfs
soft,intr,rsize=8192,wsiz=8192,auto 0 0
```

🔗 **Nota:** lo conveniente es utilizar el FQDN o el nombre de host, pero si tiene problemas para hacer ping utilice la dirección IPv4. 192.168.50+Y.3:/var/lib/one /var/lib/one nfs ...

## 2.7 Verificar los puntos de montaje

```
[root@hipervisor1 ~]# mount -s

/dev/mapper/vg_one1-lv_root on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

## 2.8 Montar el recurso exportado

```
[root@hipervisor1 ~]# mount /var/lib/one/
```

## 2.9 Compruebe que se haya montado

```
[root@hipervisor1 ~]# mount -s

/dev/mapper/vg_one1-lv_root on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

```
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
IP_del_frontend1:/var/lib/one/ on /var/lib/one type nfs
(rw,soft,intr,rsize=8192,wsiz=8192,vers=4,addr=192.168.50+Y.3,clientaddr=192.168.50+
Y.4)
```

### Paso 3 – Definir el acceso vía ssh sin password

Lo que se desea es ingresar como el usuario `oneadmin@hipervisor1` desde el equipo `frontend1`, para lograr esto el equipo `frontend1` deberá tener la llave pública del usuario `oneadmin@hipervisor1` dentro de sus claves ssh.

```
frontend1    → hipervisor1
oneadmin@frontend1 → oneadmin@hipervisor1
```

Básicamente lo que se debe realizar es:

1. Crear las claves digitales de `oneadmin@frontend1`
2. Agregar la clave pública de `oneadmin@frontend1` a `hipervisor1`
3. Configurar el equipo `hipervisor1` para aceptar accesos sin definir contraseña

#### 3.1 Ingrese como usuario `oneadmin` en `frontend1`

```
root@frontend1:~# su -l oneadmin
```

#### 3.2 Elimine las llaves digitales si existieran (opcional)

Las llaves se crean en el directorio `.ssh` dentro del directorio `home` de cada usuario.

Recuerde que **el directorio home del usuario `oneadmin` podría ser:**

- **`/home/oneadmin`**, si usted creó el usuario por comandos o con la herramienta gráfica
- **`/var/lib/one`**, si clonó el equipo `hipervisor1` a partir del `frontend1` o utiliza la MV que yo entregué.

Si no ha creado llaves digitales puede omitir este paso.

Si tiene problemas en lograr este paso elimine las claves digitales para repetirlo

Para ver si existen claves digite el siguiente comando

```
[oneadmin@frontend1: ~]$ ls -la .ssh
total 24
drwx-----. 2 oneadmin oneadmin 4096 May  8 2013 .
drwxr-x---. 12 oneadmin oneadmin 4096 Aug 22 20:19 ..
-rw-r--r--  1 oneadmin oneadmin  610 May  8 2013 authorized_keys
-rw-----  1 oneadmin oneadmin   70 Nov 20 2012 config
-rw-----  1 oneadmin oneadmin  672 May  8 2013 id_dsa
-rw-r--r--  1 oneadmin oneadmin  610 May  8 2013 id_dsa.pub
```

Las llaves digitales serán extensión `rsa`

```
oneadmin@frontend1:~$ rm .ssh/id*
```

```
oneadmin@frontend1:~$ ls -la .ssh
total 16
drwx----- 2 oneadmin oneadmin 4096 Dec  8 11:49 .
drwxr-x--- 12 oneadmin oneadmin 4096 Dec  8 11:50 ..
-rw-r--r--  1 oneadmin oneadmin  610 May  8 2013 authorized_keys
-rw-----  1 oneadmin oneadmin   70 Nov 20 2012 config
```

### 3.3 Crear las llaves digitales para el usuario oneadmin@frontend1

Para crear las llaves digitales se definirá lo siguiente:

- No se cambiará la ruta donde se guardarán las llaves
- No definirá una palabra clave

```
[oneadmin@frontend1 ~]$ ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/var/lib/one/.ssh/id_rsa): (Presione Enter)
```

```
Enter passphrase (empty for no passphrase): (Presione Enter)
```

```
Enter same passphrase again: (Presione Enter)
```

```
Your identification has been saved in /var/lib/one/.ssh/id_rsa.
```

```
Your public key has been saved in /var/lib/one/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
2a:29:52:64:0c:97:b6:4d:aa:84:81:aa:7b:9e:bb:6f oneadmin@frontend1
```

```
The key's randomart image is:
```

```
+--[ RSA 2048 ]-----+
| o ..
| o+o .
| oo+=
| o+o .
| o.. S
| o. . .
| ... o .
| ...oE.
| o==.
+-----+
```

### 3.4 Verifique que se ha creado la clave digital pública

Tal como se ha observado en el paso anterior se han creado las llaves **id\_rsa (privada)** y **id\_rsa.pub (pública)**. Guarde la llave privada y no la comparta.

El contenido de la llave digital privada se observa con el siguiente comando:

```
[oneadmin@frontend1 ~]$ cat ~/.ssh/id_rsa.pub
```

```
ssh-rsa
```

```
AAAAB3NzaC1yc2EAAAABIwAAAQEA2YA05 jdnx7WTa8ooACS0WhwSaMSKi cXKP1W1ThS4554/48HOKb5F4PP8x  
dRmHJfpuhlQIqHV2G38LRFNoBwUuE5XDuh/dlvUylobkEaKYgjYgmKlnOuHFVsVXVTwHmVIVDUOogUSgu6yN  
E9BHMew68aACY1XO7qgJIdG30ejZLdBuzpB29SrKq/6Ur6b+JGgqUGM7/i8e+DhcZbuR8eqQa18sLGSGRyAt  
eRG1QsQrhOM43XNkX0FojiooDU6V/Bdc2 juW6KJS4Qeo2ki5tSvxxXRMilKoPYkRL3/9kIuIvLLkDVSodCsIq4  
ca0EKS4d5DGCCtrcNWOqhQ7ZJah2lQ== oneadmin@frontend1
```

### 3.5 Copiar la llave pública

Este paso solo es para facilitar la comprensión que se necesitan las dos llaves públicas, y será útil si se utiliza la máquina virtual clonada del frontend1 o la provista por el instructor.

```
[oneadmin@frontend1 ~]$ cat ~/.ssh/id_rsa.pub > llave1.txt
```

### 3.6 Ingresar como usuario oneadmin en el equipo hipervisor1

```
root@hipervisor1:~# su -l oneadmin
```

Aquí hay que considerar dos situaciones para el directorio home, (en el directorio home se crea .ssh/):

- /home/oneadmin, si usted instaló una computadora con Centos y creó el usuario oneadmin por comandos o con la herramienta gráfica.
- /var/lib/one, si clonó el equipo hipervisor1 a partir del frontend1 o utiliza la MV que yo entregué.

### 3.7 Crear las llaves en el usuario oneadmin en el hipervisor1 (opcional si no estuviera creado)

```
oneadmin@hipervisor1$ ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/oneadmin/.ssh/id_rsa): (Presionar enter)
Created directory '/home/oneadmin/.ssh'.
Enter passphrase (empty for no passphrase): (Presionar enter)
Enter same passphrase again:
Your identification has been saved in /home/oneadmin/.ssh/id_rsa.
Your public key has been saved in /home/oneadmin/.ssh/id_rsa.pub.
The key fingerprint is:
54:a9:c9:07:c1:87:60:31:41:d8:82:3c:ca:b5:69:ee oneadmin@hipervisor1.empresay.com.sv
The key's randomart image is:
+--[ RSA 2048]-----+
|  . . +B=.o..      |
|  +.o...+.o       |
| ....o. ..=       |
| .. +   .+ .      |
|  o      S.        |
|  .                |
|  .                |
|  E                |
+-----+

```

### 3.8 Verifique que se han creado las credenciales

```
oneadmin@hipervisor1$ ls .ssh
id_rsa id_rsa.pub
```

```
oneadmin@hipervisor1$ cat /home/oneadmin/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaClyc2EAAAABIwAAAQEAzOEewj2bhRmMCpRgBmlqlIKkO2PsANQArCVDY/2JdL2XLX/75V1MKZJbp
LHCTpi2zCzGQdd5QyddeAUMUjikpbagUFB2IjIFRl1YtHbvZej/T+bdgbRpvPfa5lnMQsd/lXJboWrYEivdyZ
h4wYLnTvSzY8GQs4OktowMbnHlujETLsNL032zviSIfoRVIqr8ESxNkwHUMrSSJdTHIFLYCSr7V5DeoDuJRWN
cB0h1W5RaQDr+hTWA0nAUvqS1XBxQhjEtQE+6ad3ZlJFLo+2byrbMDsE7ZoWfSOnKjzYHXJn1C8CaQl07DEut
Lgr++7dZhxtRqcrvU3o6leD/VX2DRQ== oneadmin@hipervisor1
```

### 3.9 Copie la clave pública.

Esto solo es para facilitar la comprensión

```
[oneadmin@hipervisor1 ~]$ cat .ssh/id_rsa.pub > llave2.txt
```

### 3.10 Agregar las llaves públicas en el frontend1

Básicamente lo que se debe hacer es copiar el contenido de la llave1.txt y la llave2.txt al archivo ~/.ssh/authorized\_keys del equipo frontend1, así el frontend1 podrá ingresar a su propio equipo y al hipervisor1

Para realizar lo anterior se podría copiar de muchas formas (scp, filezilla, por USB, etc.), el archivo llave1.txt o llave2.txt, los cuales son una copia de los archivos id\_rsa-pub.

Si el hipervisor1 se ha instalado por aparte se deberán digitar los siguientes comandos:

```
[root@frontend1 ~]# su -l oneadmin  
  
[oneadmin@frontend1 ~]$ cat .ssh/id_rsa.pub | ssh oneadmin@hipervisor1 'cat >>  
.ssh/authorized_keys'
```

Warning: Permanently added 'hipervisor1,192.168.50.4' (RSA) to the list of known hosts.

oneadmin@hipervisor1's password: 123456

La clave id\_rsa.pub debe ser agregada en el archivo authorized-keys del servidor hipervisor1

Si el hipervisor1 ha sido clonado a partir del frontend1 o es la MV compartida por el instructor.

```
[root@frontend1 ~]# su -l oneadmin  
  
[oneadmin@frontend1 ~]$ cat llave1.txt > .ssh/authorized_keys  
  
[oneadmin@frontend1 ~]$ cat llave2.txt >> .ssh/authorized_keys
```

La clave id\_rsa.pub debe ser agregada en el archivo authorized-keys del servidor hipervisor1

### 3.11 Verifique que se haya agregado la clave de oneadmin@frontend desde hipervisor1

```
oneadmin@hipervisor1:~$ ls -l ~/.ssh/  
total 8  
-rw-r--r-- 1 oneadmin oneadmin 1222 ago 22 14:07 authorized_keys  
-rw----- 1 oneadmin oneadmin 1675 ago 22 13:47 id_rsa  
-rw-r--r-- 1 oneadmin oneadmin 418 ago 22 13:47 id_rsa.pub  
-rw-r--r-- 1 oneadmin oneadmin 391 ago 22 14:12 known_hosts  
  
oneadmin@hipervisor1:~$ cat ~/.ssh/authorized_keys  
  
ssh-rsa  
AAAAB3NzaC1yc2EAAAABIwAAAQEAzOEewj2bhRmMCpRgBmlq1IKkO2PsANQArCVDY/2JdL2XLX/75V1MKZJbp  
LHCtpi2zCzGQdd5QydeAUMUjikipbagUFB2IjIFR11YtHbvZej/T+bdgbRpvPfa51nMQsd/lXJboWrYEivdyZ  
h4wYLnTvSzY8GQs40ktowMbnHlujETLsNL032zviSIforVIqr8ESxNkwHUMrSSJdTHIFLYCSr7V5DeoDuJRWN  
cB0h1W5RaqDr+hTWA0nAUVqS1XBxQhjEtQE+6ad3ZlJFLo+2byrbMDsE7ZoWFsONKjzYHXJn1C8CaQl07DEut  
Lgr++7dZhxtRqcrvU3o6leD/VX2DRQ== oneadmin@hipervisor1  
  
ssh-rsa  
AAAAB3NzaC1yc2EAAAABIwAAAQEA2YA05jdnx7WTa8ooACS0WhwSaMSKi cXKPlW1ThS4554/48HOKb5F4PP8x  
dRmHJfpuhlQIqHV2G38LRFNoBwUuE5XDuh/dlvUylobkEaKYgjYgmKlnOuHFVsVXVTwHmVIVDUOOgUSgu6yN  
E9BHmew68aACY1XO7qgJIIdG30ejZLdBuzpB29SrKq/6Ur6b+JGgqUGM7/i8e+DhcZbuR8eqQa18sLGSGRyAt  
eRG1QsQrhOM43XNkX0FojoioDU6V/Bdc2juW6KJS4Qeo2ki5tSvxxRMilKoPYkRL3/9kIuIvLLkDVSODCsIq4  
ca0EKS4d5DGCCtrcNWOqhQ7ZJah21Q== oneadmin@frontend1
```



### 3.12 Verifique que se puede conectar desde el frontend al hipervisor1

Para que se pueda enviar comandos desde OpenNebula al hipervisor1 se necesita que se pueda ingresar directamente desde el sistema operativo del frontend al sistema operativo del hipervisor1

```
[oneadmin@one-sandbox ~]$ ssh oneadmin@hipervisor1
```

```
Warning: Permanently added 'hipervisor1,192.168.50+Y.4' (RSA) to the list of known hosts.
```

#### Paso 4 – Modificación del usuario oneadmin

En la máquina entregada por el instructor o si ha clonado el frontend1, hay que definir el directorio \$HOME del usuario oneadmin como el directorio /var/lib/one, que a su vez es el directorio que se ha exportado en el paso 1.

En el hipervisor1 se ha realizado

```
/var/lib/one ← 192.168.50+Y.3/var/lib/one (usando NFS)
```

Para el usuario oneadmin actualmente: \$HOME = /home/oneadmin, pero se cambiará a /var/lib/one Maracaná

#### 4.1 Editar el archivo /etc/passwd

Sustituir la línea oneadmin:x:9869:9869::/home/oneadmin:/bin/bash por oneadmin:x:9869:9869::/var/lib/one:/bin/bash

```
[oneadmin@hipervisor1 ~]$ mcedit /etc/passwd
```

#### 4.2 verificar que el archivo se haya modificado correctamente

```
[root@hipervisor1 ~]# cat /etc/passwd |grep one
```

```
oneadmin:x:9869:9869::/var/lib/one:/bin/bash
```

#### 4.3 Reiniciar el hipervisor1

---

Proceso para agregar el hipervisor1 (Esto lo expliqué en la clase utilizando sunstone)

Forma 1: Por comandos

Para agregar el hipervisor1 utilice la línea de comando

Ingrese como oneadmin (su -l oneadmin)

```
[oneadmin@one-sandbox ~]$ onehost create hipervisor1 --im kvm --vm qemu --net dummy
```

Forma 2: Usando herramienta Web

Ingrese como oneadmin|123456 vía web

Para agregar el hipervisor1 con sunstone, agregue el hipervisor1 y defina

IM MAD	kvm
VM MAD	qemu
VN MAD	dummy