

Docencia

Profesores de Prácticas: **Carlos Montemuiño, Sergio Villar.**

Material de las prácticas

Para las prácticas, en las máquinas Linux, la documentación será los *HOWTOs* y *manpages* del sistema. La presente guía es orientativa (no funcionará un copy paste sin saber lo que se hace).

En las prácticas se usarán máquinas virtuales (*Virtual Machines VM*) con OpenNebula. A OpenNebula se puede acceder desde todas las máquinas del laboratorio. Existen varios templates con sistema operativo Linux ya instalado.

Para acceder al entorno de prácticas se ha habilitado un acceso remoto a los servicios OpenNebula en esta dirección:

<http://nebulacaos.uab.cat:8443>

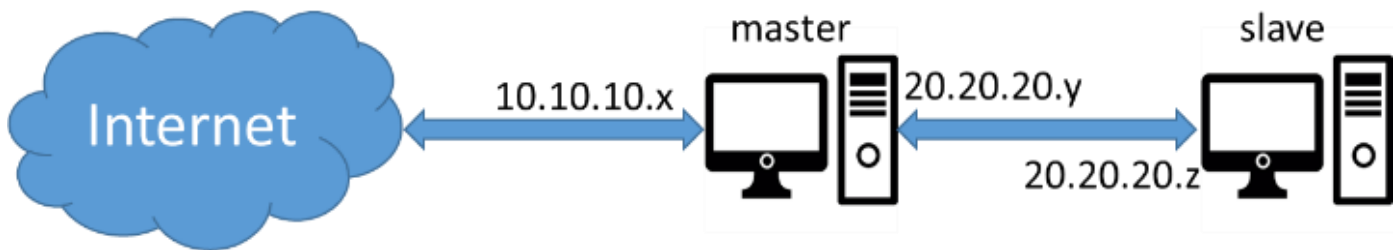
La documentación básica sobre el entorno Open Nebula y cómo utilizarlo puede consultarse en el manual de Operación: <http://docs.opennebula.org/5.4/operation/index.html>

1. Descripción del entorno

Para la realización de esta práctica diseñaremos un entorno con dos VM: (1) una máquina actuará de *Master* con dos interfaces de red y (2) la otra de *Worker* con una única interfaz.

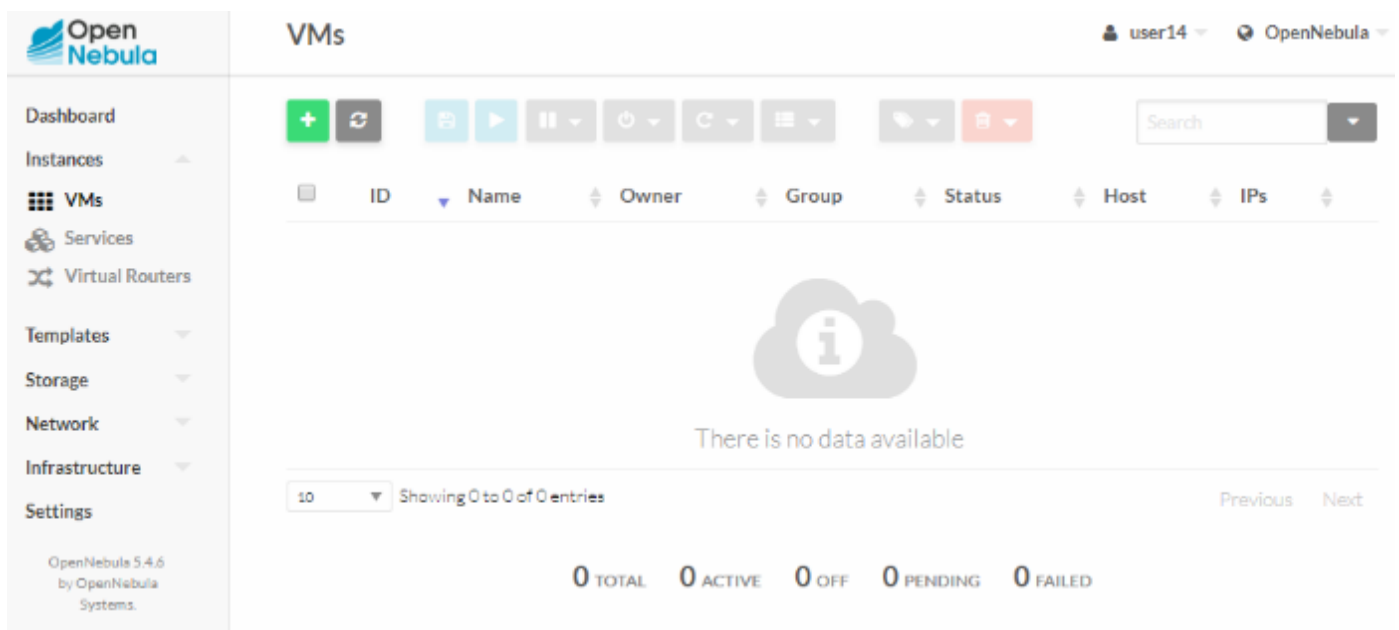
Para la máquina *Master*, se deberá tener una interfaz con acceso a Internet (Interface Internet con IP 10.10.10.0/24) y otra interfaz de red interna (Interface Middle con IP 20.20.20.0/24). En el caso de la máquina *Worker*, se debe tener una única interfaz con red interna (Interface Middle con IP 20.20.20.0/24), utilizada para comunicarse con el *Master* y obtener acceso a Internet a través de la máquina *Master*.

El entorno a crear puede verse en la siguiente imagen:



2. Creación del entorno y las máquinas virtuales

Después de acceder a OpenNebula, en el menú de la izquierda se debe seleccionar *Instances* y en el desplegable seleccionar *VMs*. Así nos aparecerá una pantalla similar a la siguiente:



En esta pantalla, se debe clicar el botón verde (+) para crear una máquina y aparecerán los diferentes templates disponibles; de la lista se debe seleccionar el template CentOS 7 - KVM - X

El nombre de la máquina virtual (VM name) será *Master* y no se deberá cambiar ningún otro campo.

En el apartado *Network* se deben añadir dos interfaces de red con el botón [Network Interface](#). La primera interfaz debe ser conectada a la red con *Id 0* y *Name Internet*. La segunda interfaz se crea con *Id 1* y *Name Middle*.

Una vez creados los interfaces, tendremos debajo del menú “Network” dos elementos:

- Interface Middle
- Interface Internet

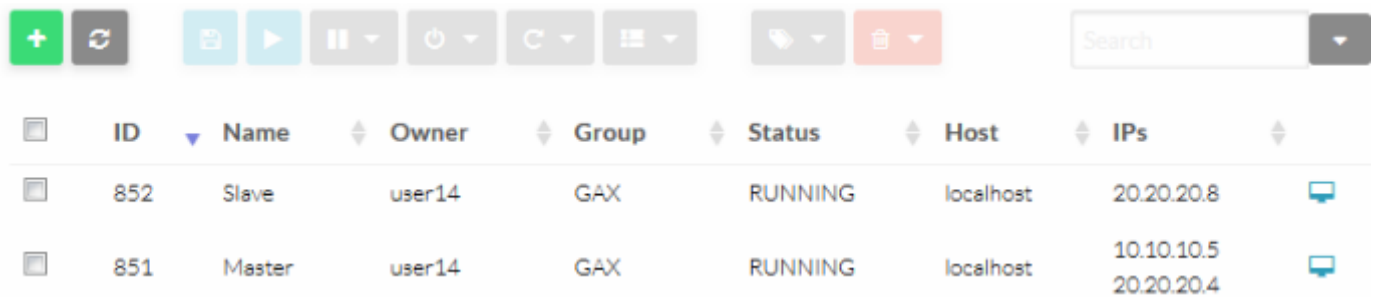
En este momento ya podemos crear la máquina virtual con el botón “CREATE” del inicio de la página. Como resultado, podemos ver la máquina Master creada en nuestra lista de VMs.

Ahora se crea la segunda máquina virtual siguiendo el mismo proceso pero con nombre *Worker* y una única interfaz de red con *Id 1* y *Name Middle*.


Al final de este paso tenéis que tener una máquina worker con una IP de rango a 20.20.20.0/24 y una máquina máster con dos direcciones IP, una en el rango 10.10.10.0/24 y otra del rango 20.20.20.0/24

3. Acceso a las máquinas virtuales y configuración de red

Una vez creadas las dos máquinas virtuales, deberíamos tener lo siguiente (con IPs diferentes):



| ID | Name | Owner | Group | Status | Host | IPs |
|-----|--------|--------|-------|---------|-----------|--------------------------|
| 852 | Slave | user14 | GAX | RUNNING | localhost | 20.20.20.8 |
| 851 | Master | user14 | GAX | RUNNING | localhost | 10.10.10.5 20.20.20.4 |

Ahora entramos a la interfaz gráfica de la máquina *Master* pulsando el icono  de la derecha. En ese momento accederemos al interface de usuario de la máquina Master para terminar de configurarla. Para entrar en la máquina se debe utilizar el usuario *root* con la contraseña *NebulaCaos* seleccionando la opción *¿No está en la lista?* para poder entrar con el usuario *root*.

Se debe cambiar el nombre de la máquina virtual dentro de la imagen. Abrimos un terminal, por ejemplo, buscando la aplicación “terminal”. Para poder poner el nombre deseado se debe modificar el fichero de configuración */etc/hostname* utilizando un editor de texto, como por ejemplo **gedit**, con el siguiente comando

```
gedit /etc/hostname
```

Y cambiar *localhost.localdomainworker* por *master*. Después del cambio se debe reiniciar la máquina para actualizar el nombre.

```
reboot
```

Después de reiniciar, abrid la consola y verificad el cambio del *hostname*.

Alternativamente podéis usar el comando:

```
hostnamectl set-hostname master
```

Y cerrar y abrir el terminal de nuevo.

Ahora se debería hacer el mismo proceso para el *Worker* cambiando el nombre de *localhost.localdomain* por *worker*. Para ello, podemos usar el botón “open in a new window”. A partir de aquí, se creará una nueva pestaña con nuestra sesión de consola en Master y la pestaña anterior reflejará la lista de nuestras VMs creadas. Es necesario que las máquinas pasen de **PENDING** a **RUNNING** para poder iniciar una sesión interactiva.

Además, en las dos máquinas se debe añadir en el fichero */etc/hosts* las IPs de las máquinas, quedando las primeras líneas del fichero similares a las siguientes tablas. Atención: fijaos bien en las direcciones de vuestras máquinas para configurar bien vuestra red.

| Master | Worker |
|---------------------|---------------------|
| 127.0.0.1 localhost | 127.0.0.1 localhost |
| ::1 localhost | ::1 localhost |
| 10.10.10.7 masterI | 20.20.20.4 worker |
| 20.20.20.4 masterM | 10.10.10.7 masterI |
| 20.20.20.8 worker | 20.20.20.8 masterM |

Para comprobar el correcto funcionamiento de la modificación de los ficheros, se debe hacer *ping* entre las dos máquinas y comprobar si las dos máquinas pueden hacer *ping* correctamente.

Desde master:

```
ping worker
```

Desde el worker:

```
ping master
```

Configuración de red

Ahora configuramos *Master* y *Worker* para conseguir acceso a Internet desde la máquina *Worker*. Para ello se utilizan ficheros de configuración para hacer los cambios permanentes. Es necesario seguir los siguientes pasos en la máquina *Master*:

Abrimos para editar el archivo */etc/sysctl.conf*, añadimos la línea *net.ipv4.ip_forward=1* para activar de forma permanente el ip forwarding y se guarda el fichero.

También debemos añadir regla de **iptables** para redireccionar los paquetes del *Worker* enviados al *Master* con el exterior de destino con el siguiente comando de terminal. Es importante que la dirección pública del Master sea la adecuada a nuestra red.

```
iptables -t nat -A POSTROUTING -s 20.20.20.0/24 -o ens3 -j SNAT --to 10.10.10.5
```

Además, Centos 7 lleva incluido un firewall (**firewall-cmd**) y debemos añadirle unas reglas al firewall para poder permitir el enmascaramiento de paquetes y el tráfico a través de la máquina *master*.

```
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -o ens3 -j MASQUERADE

firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i ens3 -o ens4 -m state
--state RELATED,ESTABLISHED -j ACCEPT
```

```
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i ens4 -o ens3 -j
ACCEPT
```

Con la finalidad de tener las reglas activadas al reiniciar la máquina, modificamos el fichero **/etc/rc.local** y añadimos las reglas al final del fichero. Y se ejecuta el comando comentado dentro del fichero.

Ahora se reinicia la máquina *Master* y, mientras esperamos, entramos a la máquina *Worker* y cambiamos la configuración de las interfaces de red con el objetivo de asignar una puerta de entrada a la máquina.

En el fichero **/etc/sysconfig/network-scripts/ifcfg-ens3** se añade el Gateway de la siguiente manera

```
gedit /etc/sysconfig/network-scripts/ifcfg-ens3
```

```
DEVICE=ens3
BOOTPROTO=static
NM_CONTROLLED=no
TYPE=Ethernet
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=20.20.20.8
GATEWAY=20.20.20.4
```

Comprobamos si funciona el cambio cargando de nuevo la interfaz de red con el comando

```
systemctl restart network
```

Ahora con el navegador Firefox en la máquina virtual intentamos consultar la página web de la UAB y deberíamos tener acceso.

Para no modificar después de cada reinicio el archivo de configuración de red, modificamos el fichero **/etc/rc.local** de la siguiente manera

```
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

echo "GATEWAY=20.20.20.4" >> /etc/sysconfig/network-scripts/ifcfg-ens3
systemctl restart network
touch /var/lock/subsys/local
```

Después, ejecutamos el comando indicado por el fichero, reiniciamos la máquina virtual y comprobamos que el acceso a internet funciona accediendo a la web de la UAB:

```
wget www.uab.cat
```

En este punto es posible que no se encuentre el comando `wget`, ya que debe ser instalado o actualizado. Si al ejecutar `yum update` no funciona, tenéis dos opciones:

- 1) Abrir directamente el navegador y comprobar que los paquetes se reciben.
- 2) En el fichero `/etc/yum.conf` añadir esta línea al final del fichero:

```
http_caching=packages
```

Después usad `yum update` y podréis instalar los paquetes que necesitéis para esta práctica y/o para las siguientes.

4. Creación de usuario con sudo

A continuación crearemos un usuario llamado *alumno* en cada máquina para comprobar más adelante si funciona la conexión ssh entre las dos máquinas.

El usuario *alumno* debe ser creado en las dos máquinas utilizando el comando

```
adduser alumno
```

Este comando nos creará los usuarios y sus correspondientes home, reduciendo así la cantidad de pasos a realizar para crear un usuario si se compara con el comando *useradd*. Le daremos una contraseña mediante:

```
passwd alumno
```

Ahora al usuario *alumno* le daremos permisos de sudo mediante el fichero `sudoers`. Abrimos el fichero `/etc/sudoers` con un editor de texto después de hacer una copia por si hay alguna modificación errónea.

```
cp /etc/sudoers /etc/sudoers.backup  
gedit /etc/sudoers
```

En este fichero añadimos la línea siguiente:

```
alumno ALL=(ALL:ALL) ALL
```

Ahora para comprobar si los cambios se han llevado a cabo correctamente, abrimos un nuevo terminal. En el nuevo terminal cambiamos de usuario e intentamos actualizar la lista de paquetes disponibles en nuestra distribución Linux.

```
su - alumno  
sudo yum update
```

Si todo se ha ejecutado correctamente, seguid los mismos pasos para la otra máquina virtual.

En el caso de ejecutarse lentamente el comando `sudo`, se debe añadir una entrada en el fichero `hosts` con `127.0.0.1 [nombremáquina]`.

5. Configuración ssh

Importante: Si en algún momento sale `Gtk-WARNING **: cannot open display:`, se debería ejecutar el comando con el usuario `root` o utilizar otro editor de texto. Después se debe volver al usuario `alumno`.

Una vez acabados los anteriores apartados de la práctica, vamos a probar la conexión ssh entre las máquinas.

SSH se encuentra instalado en la máquina virtual y no hay necesidad de volverlo a instalar.

Primero se utiliza la máquina *Master*, se cierra la sesión y se entra con el usuario *alumno*. Si se sigue con el usuario *root* se deberían modificar ficheros de configuración para permitir el ssh con *root*.

Conectamos a la máquina *Worker* utilizando el comando:

```
ssh alumno@worker
```

Debería fallar, debido a la desactivación del login mediante el uso de contraseña. Para activar la utilización de contraseña, se debería hacer lo siguiente:

```
gedit /etc/ssh/sshd_config
```

Cambiar siguiente parametro a **yes**, y guardar la modificación del fichero

```
PasswordAuthentication yes
```

Reiniciar ssh:

```
service sshd reload
```

Y si todo ha salido correctamente deberíamos lo siguiente y el nombre de máquina cambiado

```
[alumno@master ~]$ ssh alumno@worker
The authenticity of host 'worker (20.20.20.8)' can't be established.
ECDSA key fingerprint is SHA256:n9PRKF898RK7C6XNSsTN2Qam0MM7e1Xh/awtTjjiceU.
ECDSA key fingerprint is MD5:76:1f:a6:53:06:c1:c7:89:09:fd:93:f8:30:c3:9f:f2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'worker,20.20.20.8' (ECDSA) to the list of known hosts.
alumno@worker's password:
Last login: Wed Sep 19 14:25:46 2018
[alumno@worker ~]$
```

En este primer intento se ha realizado la conexión mediante contraseña. Por cuestiones de seguridad es mejor utilizar login mediante un par claves (pública y privada) en lugar de contraseña mediante

```
ssh-keygen -b 4096 -t rsa
```

Dejamos el directorio por defecto y no introducimos ningún *passphrase*.

A continuación exportamos las claves públicas del usuario para ser añadido al *authorized_key* de la máquina *Worker*, para no ser preguntados por la clave al conectar con ssh

```
ssh-copy-id alumno@worker  
ssh alumno@worker
```

Ahora se debería repetir el mismo proceso desde la máquina *Worker*.

NOTA: Recordad “parar” las máquinas virtuales **[UNDEPLOY]** al finalizar las pruebas, en caso contrario se quedan consumiendo recursos del servidor OpenNebula.