



UNIVERSIDAD  
DE GRANADA



# Cloud Computing: Servicios y Aplicaciones

---

Curso 2016 - 2017

## Guión de prácticas

### Despliegue de MVs y aplicaciones web

El **objetivo** de esta práctica es familiarizarse con el uso de una plataforma IaaS y desarrollar habilidades de despliegue de máquinas virtuales y aplicaciones web sencillas.

Para ello el alumno deberá realizar las **tareas** que se describen a continuación y entregar documentación describiendo con el mayor detalle posible todas las actividades realizadas.

### Tareas

Dentro de la plataforma de prácticas habilitada para la asignatura, *OpenNebula*, accesible a través de [docker.ugr.es](https://docker.ugr.es) (vía *ssh*), cada alumno deberá:

1. Crear dos MVs, cada una con una distribución de Linux (p.ej. CentOS, Fedora o Ubuntu).
  - a. En la primera MV instalará y configurará un servidor web (p.ej. apache, nginx, o cualquier otro que el alumno decida).
  - b. En la segunda instalará y configurará un SBGB (p. ej. MySQL, PostgreSQL, etc.).
2. Desarrollar una aplicación web sencilla alojada en la MV1, que use una base de datos manejada por el SBGB instalado en la MV2. La aplicación web debe incluir el uso de formularios y la consulta y modificación de datos almacenados en la BD.
3. Realizar el despliegue de ambas MVs, para evaluar el funcionamiento de la aplicación.
4. Elaborar un breve documento detallando el trabajo realizado.

## Opcionales

5. Crear MVs con S.O. distintos de los disponibles en el MarketPlace (p.ej. Fedora o Ubuntu).
6. Crear una MV para desplegar un servidor de *owncloud* (<https://www.owncloud.org>).

## Documentación

Es necesario entregar un informe con la siguiente estructura:

1. Portada con:
  - a. Nombre de la asignatura
  - b. Nombre de la práctica
  - c. Nombre del alumno
  - d. Dirección de correo electrónico
2. Enunciado de la práctica, descrito en un único párrafo
3. Configuración de la MV1 (con servidor web).
4. Configuración de la MV2 (con SGBD).
5. Descripción de la aplicación web. Objetivo, funcionalidad, arquitectura software, base de datos, tablas.
6. Breve manual de despliegue de las MVs.
7. Breve manual de uso de la aplicación web.

### *Opcionales*

8. Descripción del proceso de instalación del S.O. (desde una .ISO) en la MV.
9. Descripción del proceso de instalación, configuración y despliegue de Owncloud.

El documento se entregará en formato PDF.

El código fuente de la aplicación (con todos los archivos necesarios, incluidos los scripts de creación de BD y tablas), se organizará en una estructura de directorios razonable, debajo de una carpeta raíz llamada "appweb".

Tanto el código fuente como el documento, se empaquetará en un único fichero .zip con el nombre "practicaIaaS" y se entregará a través de la plataforma docente de decsai.

## Fecha límite para la entrega:

27 de Marzo de 2017 a las 23:59h.

## Apéndice A: Instalación, configuración y despliegue de servicios en dos MV interconectadas

Como tutorial para el acceso, uso y configuración de OpenNebula, se ha creado una documentación completa cuyo acceso se puede llevar a cabo desde la siguiente URL:

[https://github.com/manuparra/MasterDegreeCC\\_Practice](https://github.com/manuparra/MasterDegreeCC_Practice)

Adicionalmente, muchos de los pasos descritos en el anterior tutorial se resumen igualmente a continuación.

### 1. Introducción

Para esta práctica se usará el sistema *OpenNebula* instalado en *docker.ugr.es* y consistirá básicamente en el despliegue de dos máquinas virtuales interconectadas para dar servicio de HTTP (Apache o similar) y BBDD (MySQL o cualquier otro).

Para esta práctica se pueden elegir dos opciones de instalación, cada una con sus ventajas e inconvenientes:

- **Procedimiento A:** Instalación de máquinas virtuales (SO) preconfiguradas.
- **Procedimiento B:** Instalación de máquinas virtuales (SO) personalizadas por el usuario usando *VirtualBox* e integrándolas en *OpenNebula*.

### 2. Procedimiento A (sencillo)

Este procedimiento utiliza el MarketPlace para realizar la instalación, despliegue y configuración de los servicios.

#### Conexión al sistema

Para la conexión a *OpenNebula*, podemos usar dos mecanismos:

##### Si usas WINDOWS:

Instalación de un cliente de SSH (putty):

Descargable desde: <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Ejecutar la aplicación putty.exe

##### Si usas UNIX/LINUX:

Debes tener instalado un cliente SSH, por defecto ya tiene instalado el comando ssh.

Para conectarnos usamos nuestro login (mcc<DNI>), donde <DNI> será vuestro número de DNI (sin letra) o tarjeta de residente, y password que se asignó para cada alumno en la primera clase de *OpenNebula* (CC.2017pw):

```
ssh mcc75987461@docker.ugr.es
```

Tal y como se indica en el ejemplo, el DNI irá sin letra, pero la tarjeta de residente que lo lleva al principio sí que tendrá las letras (en mayúsculas).



### Conexión con PUTTY:

Host Name: [docker.ugr.es](https://docker.ugr.es)  
LOGIN: **mcc<DNI>**  
CLAVE: **CC.2017pw**

### Conexión desde UNIX/LINUX:

Desde un terminal (shell):

```
ssh mcc<DNI>@docker.ugr.es  
clave: CC.2017pw
```

Se recomienda modificar la contraseña una vez accedemos al sistema con el comando *"passwd"*

### Iniciar la sesión en OpenNebula.

El inicio de sesión en *OpenNebula* es obligatorio cada vez que no se ha accedido durante un largo periodo de tiempo, ya que caduca la sesión (12 horas por defecto). Para volver a generar una sesión de usuario hay que ejecutar el siguiente comando:

```
oneuser login mcc75887461 --ssh --force
```

Una vez hecho esto todos los comandos de *OpenNebula* estarán disponibles y no obtendremos el error de autenticación. Para conocer todas las órdenes de este entorno, podéis consultar el apéndice B de este documento.

### Comprobación de acceso al servicio via WEB con SunStone

Para ello debemos acceder a la siguiente web desde el navegador:

<http://docker.ugr.es:9869/>

Accedemos a esta pantalla:



**Username** usamos nuestro login: **mcc<DNI>**

El gestor **OpenNebula** utiliza otra autenticación para los servicios vía **Sunstone**, y la clave, no es la misma que se usa para ssh.

Para saber la clave que se te ha asignado:

— Entra en la sesión de SSH (con putty o ssh) con tu cuenta de SSH.

— Escribe en el terminal:

**more .one/one\_auth**

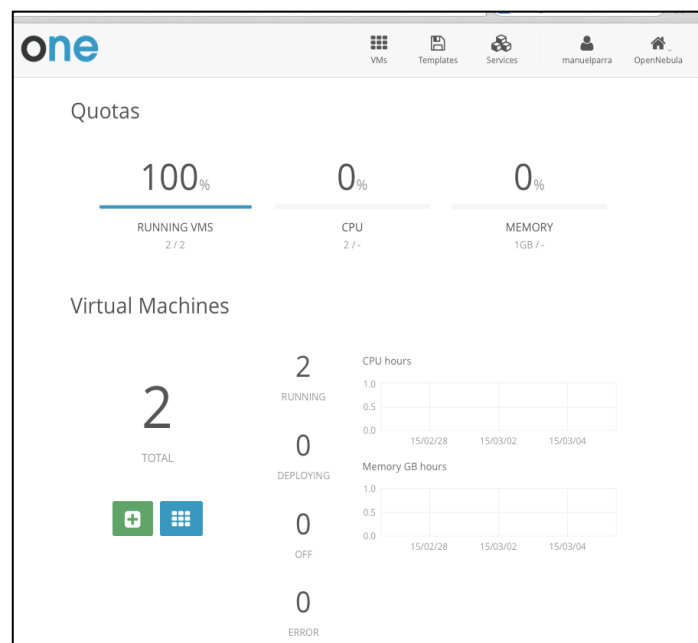
— Aparecerá nuestro login y la clave asociada similar a esto:

**mcc<DNI>:8caa31d4c2158fd42044ca2f87a3da8505272713**

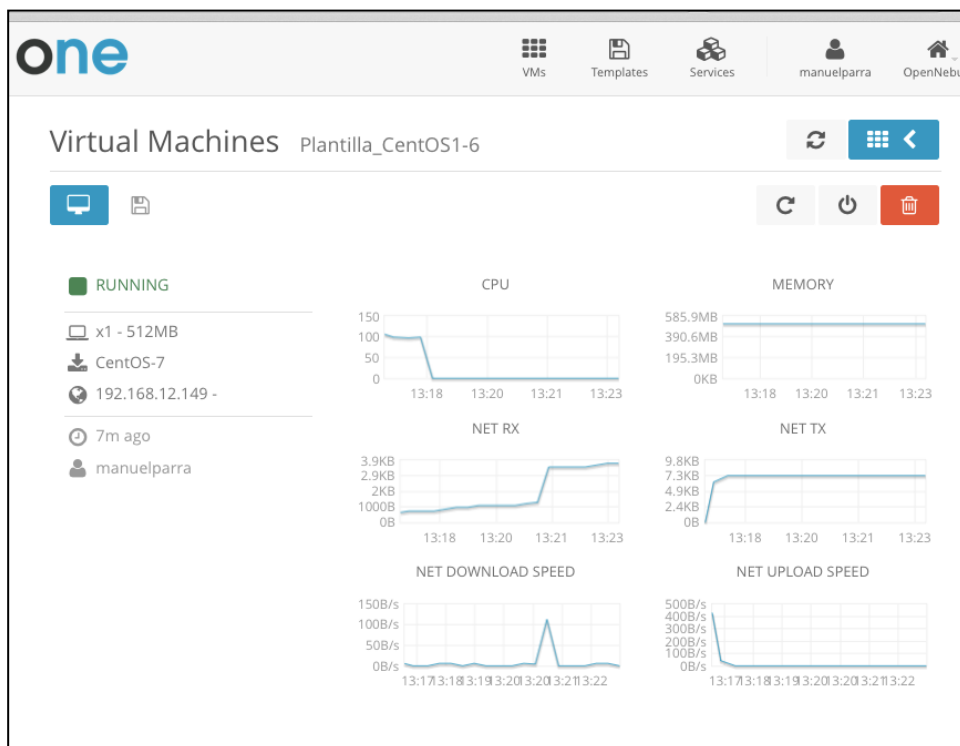
— Nuestra clave para SunStone es la cadena que hay detrás de los dos puntos

**8caa31d4c2158fd42044ca2f87a3da8505272713**

Utilizamos el nombre de usuario y clave asignados:



Todas las acciones que hemos hecho desde la línea de comandos, podemos realizarlas a través de la Interfaz web:



### Verificación de la red virtual para el usuario.

Antes de comenzar con el despliegue verificaremos que tenemos creada nuestra red virtual que nos permitirá obtener dos IP's, una para cada maquina y servicio. Ejecutamos el siguiente comando:

```
[mcc75887461@docker ~]$ onevnet list
```

En la lista que aparece, comprobamos que nuestro nombre de usuario existe y tiene en la columna LEASES un 0.

Con esto verificamos que tenemos un perfil de red creado para poder realizar las demás operaciones y que no hay asociada ninguna MV.

Lo más importante es que apuntéis vuestro ID de usuario (número de "vnet" para asociar las MVs)

### Selección de la imagen de disco a instalar.

Para ello, lo primero que haremos será comprobar cuales son las imágenes de SO, disponibles para desplegar:

```
[mcc75887461@docker ~]$ oneimage list
```

ID	USER	GROUP	NAME	DATASTORE	SIZE	TYPE	PER
STAT	RVMS						
8	oneadmin	users	CentOS-6.5-one-	default	10G	OS	No
used	1						
9	oneadmin	users	CentOS-7	default	10G	OS	No
rdy	0						
10	oneadmin	users	Ubuntu-14.04	default	10G	OS	No
rdy	0						
11	oneadmin	users	Hadoop 1.2 Mast	default	1.3G	OS	No
rdy	0						
12	oneadmin	users	Hadoop 1.2 Slav	default	1.3G	OS	No
rdy	0						

De las imágenes que aparecen vamos a utilizar CentOS 6.5 que tiene el ID asignado: **8**. Podemos utilizar cualquier otro SO, si es que estamos más familiarizados con él. Igualmente podríamos descargar cualquier otro del “Marketplace” de OpenNebula, necesario para la parte opcional de la práctica.

Como que en el caso anterior, recordaremos este ID para los siguientes pasos.

### Creación de la plantilla de Máquina Virtual para el servicio WEB

Para la maquina virtual que albergará el servicio WEB usaremos la siguiente instrucción:

Para el servicio Web crearemos una maquina virtual con un procesador y 1GB de RAM. Por favor indicad correctamente los datos de configuración para no saturar el sistema.

```
onetable create --name "P_ServicioWeb" --cpu 1 --vcpu 1 --memory 1024 --arch x86_64 --disk 8 --nic <numero de tu vnet> --vnc --ssh --net_context
```

recordad que <numero de tu vnet> corresponde con el identificador de tu red virtual (onevnet list). Igualmente es importante indicar el mismo tipo de arquitectura que tenga el SO que vamos a instalar.

Explicamos los parámetros:

- - *name* : INDICA EL NOMBRE DEL plantilla a crear
- - *cpu* : El numero de CPUS que vamos a utilizar
- - *memory* : La memoria que se usara para la Maquina Virtual
- - *arch*: La arquitectura que se va a usar
- - *disk*: El ID del DISCO/IMAGEN que vamos a usar . REVISAR: **oneimage list**

- - *nic* : El nombre de la RedVirtual que usaremos. REVISAR el ID o Nombre: **onevnet list**

- - *arch*: La arquitectura que se va a usar.

- - *ssh*: acceso via ssh

- - *net\_context*: Usaremos el contextos para la imagen

### Creación de la plantilla de Maquina Virtual para el servicio MySQL

Para el servicio Web crearemos igualmente una maquina virtual con un procesador y 1GB de RAM. Basta con seguir el mismo procedimiento, dando otro nombre identificativo. Por tanto usaremos la siguiente instrucción:

```
onemplate create --name "P_ServicioMySQL" --cpu 1 --vcpu 1 --memory 1024 --arch x86_64 --disk 8 --nic <numero de tu vnet> --vnc --ssh --net_context
```

donde de nuevo <numero de tu vnet> corresponde con el identificador de tu red virtual (onevnet list)

### Verificamos que están creados los templates y están listos para ser instanciados.

Para ello usamos el comando:

```
[mcc75887461@docker ~]$ onemplate list
```

Veremos que aparecen los dos *templates* (plantillas) que hemos creado. La lista aparecerá similar a esta:

ID	USER	GROUP	NAME	REGTIME
14	test_user	users	Plantilla_CentOS	03/06 11:44:47

Ahora para el siguiente paso usaremos el código para instanciar la MV.

### Instanciamos los templates recién creados.

En el comando anterior aparecen los dos templates, cada uno con su ID. Para instanciar cada uno de los templates usamos el siguiente comando:

Para el servidor Web:

```
[mcc75887461@docker ~]$ onemplate instantiate <id template>
```

Para el servidor MySQL:

```
[mcc75887461@docker ~]$ onemplate instantiate <id template>
```



Una vez hecho esto verificamos que ambas maquinas estén listas, comprobando su estado:

```
[mcc75887461@docker ~]$ onevm list
```

Esperamos un rato hasta que el estado (STAT) indique: **runn**

Debido a que el arranque de la maquina virtual lleva su tiempo, veremos el estado a cada momento:

ID	USER	GROUP	NAME	STAT	UCPU	UMEM	HOST
TIME							
5	manuelpa	users	Plantilla_Cent0	runn	0	512M	noded07
0d 01h13							

En la columna STAT tenemos el estado actual. RUNNING = RUNN  
Se obtiene información sobre la Memoria, HOST, etc.

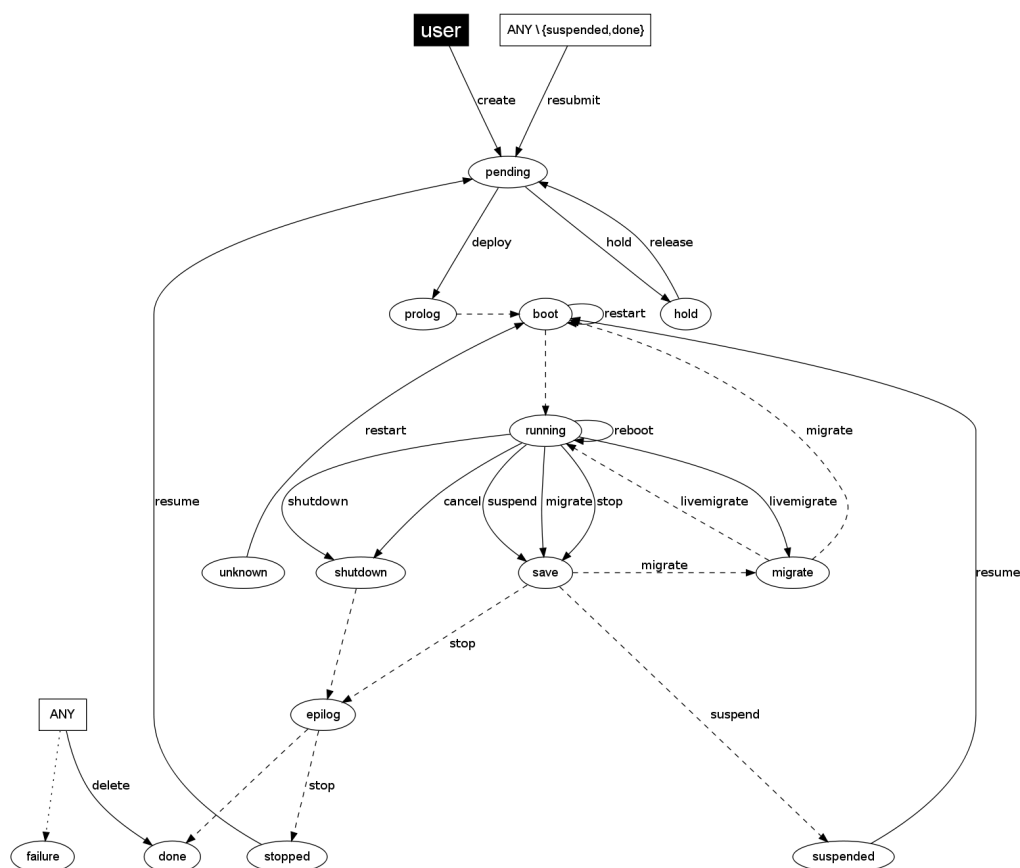


Figura 1. Mapa de estados de las máquinas virtuales

### Instalación del servicio web en la primera maquina virtual creada

Para realizar la instalación en la primera maquina virtual, lo que hacemos será comprobar la IP.

```
[mcc75887461@docker ~]$ onevm show <id_maquina_virtual servidor web>
```

Nos interesa saber la IP que se le ha asignando a la MV del servicio WEB.

En el listado que aparece buscamos donde indica: **ETH0\_IP**

Esta IP será la que se ha asignado a la MV para el servicio web.

El comando anterior nos muestra la información detallada de la maquina Virtual en funcionamiento.

```
... .  
VIRTUAL MACHINE 5 INFORMATION  
ID : 5  
NAME : Plantilla_CentOS-5  
USER : manuelparra  
GROUP : users  
STATE : ACTIVE  
LCM_STATE : RUNNING  
RESCHED : No  
HOST : noded07  
CLUSTER ID : -1  
CLUSTER : default  
START TIME : 03/06 11:45:00  
END TIME : -  
... .
```

La parte que nos interesa de los detalles es saber que IP se le ha asignado a esta maquina creada:

Vamos a la parte que indica:

```
CONTEXT=[  
  DISK_ID="1",  
  ETH0_DNS="150.214.191.10",  
  ETH0_GATEWAY="192.168.12.1",  
  ETH0_IP="192.168.12.148",  
  ETH0_MAC="02:00:c0:a8:0c:94",  
  NETWORK="YES",
```

ETHO\_IP: **"192.168.12.148"**,

Una vez localizada la IP y la máquina corriendo:

Si no aparece esta información, puede que hayamos creado mal la plantilla de máquina virtual, por lo que conviene borrarla y volver a crearla. Las órdenes asociadas se pueden encontrar en el apéndice B.

Si todo va bien, accedemos a la maquina desde el SHELL / TERMINAL usando:

```
[mcc75887461@docker ~]$ ssh root@<ip_asignada servidor web>
```

Nos aparece un mensaje de validación de nuestra KEY, aceptamos (Y)

Una vez dentro del sistema actualizamos el sistema operativo (conveniente para tenerlo todo en orden):

```
[mcc75887461@docker ~]$ yum -y update
```

Una vez que termine de actualizar todo, reiniciamos la MV:

```
[mcc75887461@docker ~]$ reboot
```

La MV se cerrará y reiniciará. Esperamos a que esté activa comprobando su estado (normalmente tardará poco en este caso):

```
[mcc75887461@docker ~]$ onevm list
```

Una vez que esté RUNNING, volvemos a entrar:

```
[mcc75887461@docker ~]$ ssh root@<ip_asignada servidor web>
```

Podemos comprobar que tenemos acceso al Exterior:

```
ping -c2 google.com
```

Ahora ya tenemos la maquina virtual disponible para poder desplegar cualquier servicio. Ahora se deben instalar todos los servicios web. Este apartado de la práctica dependerá del SO seleccionado y por tanto se deja a expensas del alumno.

### Instalación del servicio MySQL en la segunda maquina virtual creada

Para realizar la instalación en esta segunda maquina virtual, realizaremos básicamente los mismos pasos que en el caso anterior. En primer lugar, lo que hacemos será comprobar la IP.

```
[mcc75887461@docker ~]$ onevm show <id_maquina_virtual servidor mysql>
```

Nos interesa saber la IP que se le ha asignando a la MV del servicio MySQL.

En el listado que aparece buscamos donde indica: **ETH0\_IP**

Esta IP será la que se ha asignado a la MV para el servicio web.

Accedemos a la maquina:

```
[mcc75887461@docker ~]$ ssh root@<ip_asignada servidor web>
```

Nos aparece un mensaje de validación de nuestra KEY, aceptamos (Y)

Una vez dentro del sistema actualizamos el sistema operativo:

```
[mcc75887461@docker ~]$ yum -y update
```

Una vez que termine de actualizar todo, reiniciamos la MV:

```
[mcc75887461@docker ~]$ reboot
```

La MV se cerrará y reiniciará. Esperamos a que esté activa comprobando su estado:

```
[mcc75887461@docker ~]$ onevm list
```

Una vez que esté RUNNING, volvemos a entrar:

```
[mcc75887461@docker ~]$ ssh root@<ip_asignada servidor web>
```

Ahora se deben instalar todos los servicios MySQL (o cualquier SGBD elegido). De nuevo es cuestión del alumno poner en marcha todo el procedimiento de acuerdo al SO y el SGBD seleccionado.

Recordad que debéis enlazar ambas máquinas virtuales, por lo que será necesario establecer permisos de acceso, reglas de cortafuegos, etc. Consultad los manuales de la distribución seleccionada para facilitar la tarea.

### 3. Procedimiento B (más complejo)

El procedimiento B se realiza mediante una combinación de *VirtualBox* y *OpenNebula*, y permite utilizar y reutilizar maquinas virtuales que el usuario ya tenga preparadas o que quiera personalizar ilimitadamente sin tener las restricciones de usar las que ya vienen en MarketPlace de OpenNebula.

#### Instalación y configuración de imágenes ISO de Sistemas Operativos dentro de OpenNebula

Otra de las opciones que permiten instalar un SO dentro de OpenNebula es por medio de una ISO concreta de un sistema operativo.

La forma habitual de usar OpenNebula para instalar o desplegar un SO, se realiza mediante el uso del MarketPlace. Esta opción es muy útil si tenemos la versión concreta que queremos instalar, pero en el caso que queramos instalar un sistema operativo o una versión de sistema operativo que no aparece en el listado de MarketPlace de OpenNebula debemos instalar el sistema operativo por otros medios.

Básicamente existen dos medios:

- Instalación desde el propio OpenNebula, usando VNC.
- Instalación mixta con Virtual Box y la contextualización de OpenNebula para MV.

En nuestro caso usaremos la segunda opción pues es mas sencilla de desplegar con un perfil de usuario.

#### Descarga de una imagen mínima ISO de un sistema operativo.

Si quieres usar Debian:

<http://cdimage.debian.org/debian-cd/7.8.0/amd64/iso-cd/debian-7.8.0-amd64-netinst.iso>

Si quieres usar CentOS:

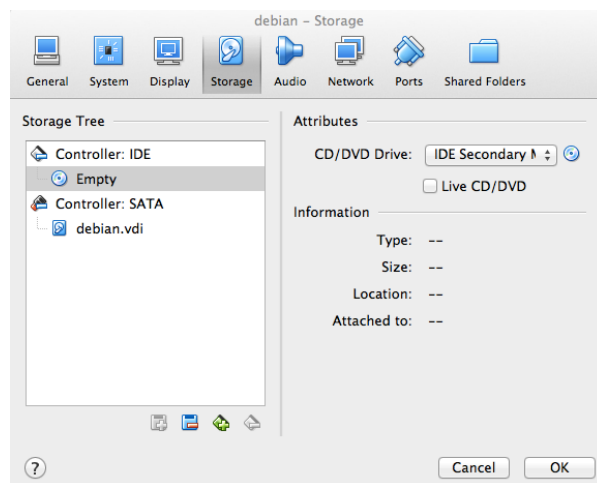
[http://centos.cadt.com/6.6/isos/x86\\_64/CentOS-6.6-x86\\_64-minimal.iso](http://centos.cadt.com/6.6/isos/x86_64/CentOS-6.6-x86_64-minimal.iso)

#### Instalación sobre VIRTUALBOX

Una vez descargada la imagen seleccionada, instalaremos VirtualBox para crear una maquina virtual completa e instalada desde la ISO que acabamos de descargar.



Para ello seleccionamos como DVD/CD de arranque la ISO que hemos descargado:



Para esta instalación inicial, no modificamos nada en el apartado Network, pues necesitamos que no haya una configuración fijada para ello.

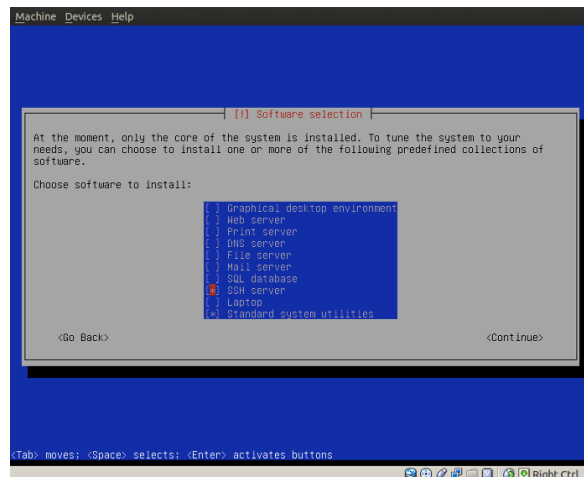
Las características de la Maquina Virtual a crear son las siguientes:

RAM: 1GB  
 HD: 8GB  
 Display: Todo por defecto  
 System: 1 core  
 Audio: Todo por defecto

Ejecutar la maquina Virtual y comenzar la instalación.

Hay que seguir todos los pasos de la instalación seleccionada (Debian o CentOS).

Para el caso de Debian (mínima) hay que recordar que es necesario instalar en los servicios SSH para que luego el acceso a la maquina virtual desde OpenNebula se pueda realizar. No es necesario el entorno Grafico a no ser que queramos usar X11 forwarding.



Una vez finalizada la instalación hay que reiniciar la maquina virtual e instalar todos los paquetes de actualización que correspondan (`yum -y update` [para CentOS] ó `apt-get upgrade` [para Debian])

### Uso del paquete de contextualización de la maquina virtual para VirtualBox

Cuando se instala la máquina virtual y se ha completado todo el proceso de instalación y actualización, es cuando se realiza la operación de contextualización de la maquina virtual.

Esta operación consiste en instalar unos servicios/plugins dentro de la maquina virtual (SO) que nos permitirán luego poder acceder, usar y gestionar la máquina desde OpenNebula. Sin estos paquetes la maquina que hemos creado no servirá para nada en OpenNebula.

Para instalar los paquetes tenemos dos opciones según el sistema de gestión de paquetes del sistema operativo:

- Según la distribución:
  - Compatible with Ubuntu 11.10 to 14.04 and Debian 6/7  
[http://dev.opennebula.org/attachments/download/847/one-context\\_4.10.0.deb](http://dev.opennebula.org/attachments/download/847/one-context_4.10.0.deb)
  - Compatible with CentOS and RHEL 6/7  
[http://dev.opennebula.org/attachments/download/846/one-context\\_4.10.0.rpm](http://dev.opennebula.org/attachments/download/846/one-context_4.10.0.rpm)

Para instalar estos servicios, es necesario descargarlo con *wget*. Si no están instalados, usar el gestor de paquetes para ello.

Hecho esto se instala:

```
Debian:    dpkg                                -i                                one-context_4.10.0.deb
CentOS:    rpm -i one-context_4.10.0.deb
```

Al hacer esto se instalan en el arranque de la maquina.

Finalmente se apaga la máquina de forma CORRECTA.

### Conversión de la imagen de disco a formato OpenNebula

OpenNebula funciona con imágenes de SO en formatos variados, pero no acepta directamente el formato de VirtualBox.

Para hacer compatible la imagen debemos realizar la conversión de la imagen de VirtualBox a QCOW2.

La operación a realizar es la siguiente:

Primero a formato **img** desde nuestro PC:

```
VBoxManage clonehd --format RAW vm.vdi vm.img
```

Segundo subir la imagen a docker.ugr.es comprimida

```
scp vm.img.zip mcc75887461@docker.ugr.es:/home/mcc75887461/
```

Tercero, descomprimir la imagen en docker.ugr.es

```
unzip vm.img.zip
```

Por último convertir a formato KVM QCOW2 dentro de la cuenta del SERVIDOR :

```
qemu-img convert -f raw vm.img -O qcow2 vm.qcow2
```

(mas info: [http://docs.openstack.org/image-guide/content/ch\\_converting.html](http://docs.openstack.org/image-guide/content/ch_converting.html) y [http://docs.openstack.org/image-guide/content/ch\\_converting.html](http://docs.openstack.org/image-guide/content/ch_converting.html))

### Conexión al sistema

Para la conexión a OpenNebula una vez realizada la conversión, podemos usar dos mecanismos Putty (Windows) o desde una shell de Linux el comando ssh.

Para conectarnos usamos nuestro login (mcc<DNI>) y password (CC.2016 por defecto) que se asignó para cada alumno en la primera clase de OpenNebula:

```
ssh mcc<DNI>@docker.ugr.es
```



### Iniciar la sesión en OpenNebula.

El inicio de sesión en openNebula es obligatorio cada vez que no se usa opennebula durante un largo periodo de tiempo. Para volver a generar una sesión de usuario hay que ejecutar el siguiente comando:

```
oneuser login mcc<DNI> --ssh --force
```

Una vez hecho esto todos los comandos de OpenNebula estarán disponibles y no obtendremos el error de autenticación.

Cada sesión dura por defecto 12 horas.

### Verificación de la red virtual para el usuario.

Antes de comenzar con el despliegue verificaremos que tenemos creada nuestra red virtual que nos permitirá obtener dos IP's, una para cada maquina y servicio.

Ejecutamos el siguiente comando:

```
[mcc75887461@docker ~]$ onevnet list
```

En la lista que aparece, comprobamos que nuestro nombre de usuario existe y tiene en la columna LEASES un 0.

Con esto verificamos que tenemos un perfil de red creado para poder realizar las demás operaciones.

### Importación de la imagen a OpenNebula

Una vez convertida la imagen a formato correcto y habiéndonos autenticado desde OpenNebula, pasamos a realizar la importación de la imagen a nuestra cuenta de OpenNebula.

Para ello usamos el comando siguiente:

```
oneimage create -d default --name MiDebian --path /home/mcc75887461/vm.qcow2 --type OS --driver qcow2 --description "MiDebian"
```

Cruzamos los dedos y si todo va bien habrá importado nuestra nueva imagen del Sistema Operativo Debian a OpenNebula.

A partir de aquí los pasos a realizar se encuentran en la sección **Selección de la imagen de disco a instalar, del PROCEDIMIENTO A**

### Instalación y configuración de la imagen preparada.

Para continuar con este Paso, hay que continuar en la Sección **PROCEDIMIENTO A: Selección de la imagen de disco a Instalar**

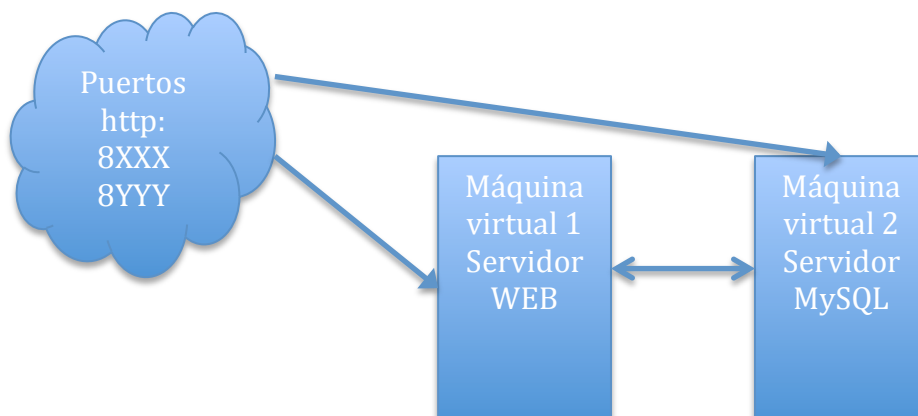
#### 4. Esquema de Interconexión de ambas máquinas virtuales y conexión con el desde fuera

Para la interconexión de ambas maquinas virtuales, hay que tener claro que cada usuario tiene asignadas dos IP's, una para el servidor WEB y otra para el servidor MySQL.

Determinados servicios como MySQL generalmente sólo permiten conexiones locales, pero en este caso como indica el diagrama, el servidor de MySQL debe permitir acceder desde la máquina del servidor WEB únicamente.

Para acceder al servicio WEB se han habilitado dos puertos por cada alumno, que corresponden a la IP de la maquina virtual precedido de 8, de modo que si la máquina virtual donde esta el servicio web es 192.168.10.123, el puerto que usaremos para el acceso al servicio web desde fuera será: 8123,

por tanto para acceder al servidor web usamos:  
<http://docker.ugr.es:8123>



## Apéndice B: Órdenes OpenNebula

onemarket	onevm		
oneacct	onedatastore	onegroup	onetemplate
onevnet			
oneacl	onedb	onehost	oneuser
onezone			
onecluster	oneimage	onevcenter	

### onemarket:

Permite listar las imágenes de Sistemas operativos, datos, etc. que podemos importar para añadir a nuestra nube disponibles desde un MarketPlace

Lista de los SO disponibles para funcionar:

```
onemarket list
```

### oneimage:

Permite listar las imágenes que están en el sistema listas para ser usadas para un despliegue de Maquinas virtuales

Lista de los imágenes importadas disponibles para funcionar:

```
oneimage list
```

### onevnet:

Permite gestionar las redes virtuales que se han creado.

Lista de las redes virtuales creadas

```
onevnet list
```

### onetemplate:

Permite gestionar plantillas de generación de maquinas virtuales, de modo que podemos crear plantillas que tenga determinados atributos de memoria, capacidad, etc. etc.

Lista de templates disponibles

```
onetemplate list
```

Es normal que aparezca vacío ya que aun no hemos realizado ninguna plantilla. La crearemos más adelante.

## **onevm:**

Permite gestionar las maquinas virtuales lanzadas, conocer el estado, modificar el estado, conocer la IP/Red, recursos, etc, etc.

Lista de maquinas virtuales disponibles:

```
onevm list
```

Es normal que aparezca vacío ya que aun no hemos lanzado ninguna maquina virtual. La crearemos más adelante.

## **onehost:**

Permite gestionar los host y recursos de la nube creada con OpenNebula

Lista de hosts del sistema y su estado

```
onehost list
```

Aparece vacío ya que las HOSTS se gestionan desde un perfil de usuario con privilegios.

## **Comandos útiles para la gestión de las maquinas virtuales**

**onevm delete ID    -> Borra una maquina Virtual**

**onevm reboot ID    -> Reinicia una maquina Virtual**

**onevm poweroff ID   -> Apaga una maquina Virtual**

**onetemplate delete ID       -> Borra una plantilla de MV**