

# OpenNebula

Introducción al uso de Cloud Computing

# Introducción

- OpenNebula es una aplicación de Cloud Computing que permite la organización, manejo y despliegue de tu propia nube.
- Es apto tanto para entornos empresariales como para entornos más sencillos debido a sus posibilidades de personalización.
- Además, es un proyecto relativamente nuevo ya que comenzó en 2005, precisamente por un equipo de españoles como trabajo de investigación.

# Entorno

- Este tutorial está escrito usando el siguiente entorno:
  - Hardware: 3.3 Ghz Intel Core I7, 16GB DDR3
  - Sistema Operativo anfitrión: Windows 8.1 (es indiferente, usamos VMs)
  - Entorno de ejecución:
    - VMWare Workstation 11.1
    - Sistema Operativo: Ubuntu 14.04 LTS Server
    - Número de máquinas virtuales: 3 (servidor, nodo y datastore)
    - OpenNebula: 4.12

# Instalación

- En este tutorial no hablaremos de la instalación para poder centrarnos más en explicar el uso básico de OpenNebula, y qué utilidades puede tener.
- Para un comienzo rápido es recomendable seguir la guía para ubuntu y KVM.
  - En esa guía se crea un entorno en el que existirán tres roles: servidor, nodo y datastore.
  - Cada rol puede ser adoptado por una máquina distinta, aunque una misma máquina podría ejecutar todos los roles.
  - Esto último no es recomendable puesto que en el mundo real no se va a tener una máquina para todo.
- Se puede seguir la guía de instalación aquí:
  - [http://docs.opennebula.org/4.12/design\\_and\\_installation/quick\\_starts/qs\\_ubuntu\\_kvm.html](http://docs.opennebula.org/4.12/design_and_installation/quick_starts/qs_ubuntu_kvm.html)
  - [http://docs.opennebula.org/4.12/design\\_and\\_installation/building\\_your\\_cloud/ignc.html](http://docs.opennebula.org/4.12/design_and_installation/building_your_cloud/ignc.html)
- Dicho documento también está en la Web docente de la asignatura.

# Sunstone, la interfaz web de OpenNebula

- La forma más sencilla de empaparse del funcionamiento de OpenNebula es mediante su interfaz web.
- Proporciona una visión global de lo que se puede hacer, además de un marketplace con un catálogo de imágenes para máquinas virtuales listas para ser usadas:
  - Sistemas Operativos completos
  - Imágenes ligeras destinadas únicamente a hacer pruebas.

# Sunstone, la interfaz web de OpenNebula

OpenNebula  
Sunstone

Dashboard

oneadmin

OpenNebula

Dashboard

System

Virtual Resources

Infrastructure

Marketplace

OneFlow

Support

Virtual Machines

15 TOTAL  
13 ACTIVE  
0 PENDING  
0 FAILED

34%

REAL CPU USAGE  
483 / 14

60%

REAL MEMORY USAGE  
7.6GB / 12.5GB

Hosts

2 TOTAL  
2 ON  
0 OFF  
0 ERROR

100%

ALLOCATED CPU  
1600 / 1600

45%

ALLOCATED MEMORY  
14.5GB / 32GB

39%

REAL CPU  
626 / 1600

17%

REAL MEMORY  
5.5GB / 32GB

Storage

5 IMAGES  
200MB USED

Users

6 USERS  
3 GROUPS

Network

2 VNETS  
16 USED IPS

OpenNebula 4.8.0 by C12G Labs.

# Almacenamiento

- El almacenamiento en OpenNebula se produce en los datastores.
- Estos datastores sirven para almacenar las imágenes de las máquinas virtuales. Pueden ser de tres clases:
  - Imágenes: albergan el repositorio de imágenes de disco, es decir, las imágenes que podemos usar para crear nuevas máquinas virtuales. Desde este datastore se copian o mueven las imágenes cuando se crea una máquina.
  - Sistema: en este tipo se encuentran las máquinas virtuales en ejecución.
  - Archivos: un datastore especial que no alberga imágenes. Puede albergar cualquier otro tipo de archivo para ser usado posteriormente, como por ejemplo un archivo de contexto para pasarlo a una máquina virtual.

# Almacenamiento (2)

- Por defecto OpenNebula se instala con tres datastores, uno de cada tipo de los mencionados: default, system y files, siguiendo el orden en el que los hemos descrito.
- Lo normal es instalar datastores en servidores NAS que, en función de la tecnología que tengan por debajo, la transferencia de imágenes a un nodo puede ocurrir de distintas formas, según compatibilidad y drivers.
- OpenNebula es compatible con varios, incluyendo la opción de un directorio compartido (NFS) o copiado de las imágenes a través de ssh.
- También existen drivers más específicos como qcow2 o vmfs, de qemu y VMware respectivamente, etc.
- Para más información sobre el almacenamiento consultar el siguiente enlace:  
<http://docs.opennebula.org/4.12/administration/storage/index.html>



# Gestión típica. Operaciones básicas: Imágenes

- El primer paso para usar OpenNebula es crear o instalar una imagen en el servicio, que será normalmente un sistema operativo listo para arrancarse.
- Recordamos que existe la posibilidad también de instalar una imagen a través del marketplace disponible a través de la interfaz web de OpenNebula.
- Nosotros vamos a usar el comando *oneimage* para instalar nuestra propia imagen. Podemos hacerlo a través de una plantilla o a través de línea de órdenes.
- Al crear una imagen hay dos elementos indispensables:
  - El nombre de la imagen a crear y
  - La ubicación del archivo (la imagen de disco).
- Se pueden especificar otros elementos como el datastore específico en el que guardar la imagen, si va a ser persistente o no (esto permitirá arrancar la máquina virtual casi al instante en lugar de tener que hacer una copia de la imagen), el driver que se va a usar (qcow2, raw, etc.), etc.

```
$> oneimage create --name "imagen ubuntu" --driver qcow2 \  
--path "/home/local/imagenes/ubuntu_14.04_desktop.qcow2" \  
--type OS --datastore default
```

- A través de esta orden podemos también eliminar cualquier imagen existente e incluso clonaras, entre otras cosas.



# Redes virtuales

- El segundo paso es crear una red virtual que asignar a nuestras máquinas virtuales (aunque es posible desplegar máquinas virtuales sin red).
- Estas redes mapean una o varias redes el nodo en el que se despliega la máquina virtual, por tanto, dependen de la configuración de red en el nodo.
- Para las redes hay varios elementos que configurar:
  - Atributos clave, como el nombre de la red, dispositivo bridge, si se activa VLAN, etc.
  - Espacio de direcciones disponible (Address Range, AR): tipo de red (IP4, IP6, Ethernet), tamaño de la red, rango de direcciones, etc.
  - Parámetros de contextualización, como puerta de enlace, servidor DNS, máscara de red, etc.
  - Es recomendable crear las redes a partir de plantillas debido al gran número de atributos que es necesario definir. Por tanto, creamos la plantilla y la instalamos con el comando *onevnet*.
  - Se puede comprobar con el siguiente ejemplo cómo quedaría una red sencilla.



# Redes Virtuales



```
1 NAME = "red"
2 BRIDGE = br0
3
4 #Parámetros de context
5 NETWORK_ADDRESS = "192.168.1.0"
6 NETWORK_MASK    = "255.255.255.0"
7 DNS             = "8.8.8.8"
8 GATEWAY         = "192.168.1.1"
9
10 #Rango de direcciones
11 AR=[
12     TYPE = "IP4",
13     IP   = "192.168.1.100",
14     SIZE = "10"
15 ]
```

Redes Virtuales

# Redes Virtuales

- Nosotros tenemos por defecto asignada una red virtual con 2 Ips
- Haremos uso de dichas redes para instanciar nuestras máquinas virtuales

Select cluster  

<input type="checkbox"/>	ID ▼	Owner	Group	Name	Reservation	Cluster	Leases
<input type="checkbox"/>	230	ahilario	users	ahilario_vnet	No	-	0 / 2
<input type="checkbox"/>	229	mcc76654141	oneadmin	mcc76654141_vnet	No	-	0 / 2
<input type="checkbox"/>	228	mcc76439773	oneadmin	mcc76439773_vnet	No	-	0 / 2
<input type="checkbox"/>	227	mcc76139799	oneadmin	mcc76139799_vnet	No	-	0 / 2
<input type="checkbox"/>	226	mcc75933306	oneadmin	mcc75933306_vnet	No	-	0 / 2
<input type="checkbox"/>	225	mcc75928532	oneadmin	mcc75928532_vnet	No	-	0 / 2
<input type="checkbox"/>	224	mcc75928287	oneadmin	mcc75928287_vnet	No	-	0 / 2
<input type="checkbox"/>	223	mcc75926571	oneadmin	mcc75926571_vnet	No	-	0 / 2
<input type="checkbox"/>	222	mcc75575731	oneadmin	mcc75575731_vnet	No	-	0 / 2
<input type="checkbox"/>	221	mcc75571687	oneadmin	mcc75571687_vnet	No	-	0 / 2

Showing 1 to 10 of 103 entries

Previous **1** 2 3 4 5 ... 11 Next

10 ▼

# Plantillas para máquinas virtuales

- Para poder usar una imagen de nuestro repositorio y, en definitiva, crear una nueva máquina virtual es necesario especificar una plantilla de la misma. Esto se consigue a través de la orden *onetemplate*.
- Las plantillas para máquinas virtuales necesitan varios elementos para ser instanciadas:
  - Asignación de recursos de la máquina virtual, como por ejemplo la cantidad de memoria RAM o el número de CPUs de los que va a disponer.
  - La imagen que va a usar, de las que hemos creado con anterioridad.
  - Pertenencia a alguna red virtual, como la creada en el apartado anterior. Aunque es opcional suele estar muy presente: interesa que nuestras máquinas virtuales.
  - Sumándose a éstos hay otros atributos muy interesantes que, aunque opcionales, pueden ser de gran utilidad:
    - Opciones de contextualización. Abren la puerta a todo tipo de personalizaciones para nuestra máquina.
    - Posibilidad de añadir un servidor VNC a la máquina para después conectarnos remotamente.

# Plantillas para máquinas virtuales

- Un ejemplo de creación de una imagen usando la red de ejemplo y la imagen de ejemplo creadas en los apartados anteriores sería así en línea de órdenes, usando el comando *onetemplate*:

```
$> onetemplate create --name "Ubuntu template" --cpu 1 --vcpu 1 --arch x68_64 \
--memory 2048 --disk "imagen ubuntu" --vnc --nic "red" --ssh
```

- Como podemos observar el número de argumentos es demasiado grande.
- Del mismo modo que antes con la red, se puede crear la plantilla en texto plano y luego agregarla al sistema mediante la orden anterior.
- Por supuesto es posible actualizar o eliminar también las plantillas usando la misma orden.

# Máquinas virtuales, al fin

- Llegamos por fin a lo que pretendíamos desde el principio: crear máquinas virtuales.
- Afortunadamente esta fase es muy sencilla, únicamente hay que instanciar una de las plantillas de máquinas virtuales que hemos creado y listo.
- Se pueden listar las plantillas que tenemos antes, e instanciar nuestra máquina por medio del nombre o el ID.

```
$> onetemplate list
```

ID	USER	GROUP	NAME	REGTIME
6	oneadmin	oneadmin	Ubuntu template	09/28 17:20:07

```
$> onetemplate instantiate "Ubuntu template"
```

```
$> onetemplate instantiate 6
```



# Máquinas virtuales, al fin

- Una vez hecho esto, el planificador de OpenNebula será el encargado de asignar a un nodo que satisfaga los requisitos de la plantilla el despliegue de la máquina virtual.
- Podremos ver el estado de las máquinas virtuales con *onevm list*.

```
$> onevm list
```

ID	USER	GROUP	NAME	STAT	CPU	MEM	HOSTNAME	TIME
0	oneadmin	oneadmin	my_vm	pend	0	0K		00 00:00:03

# Contextualizar

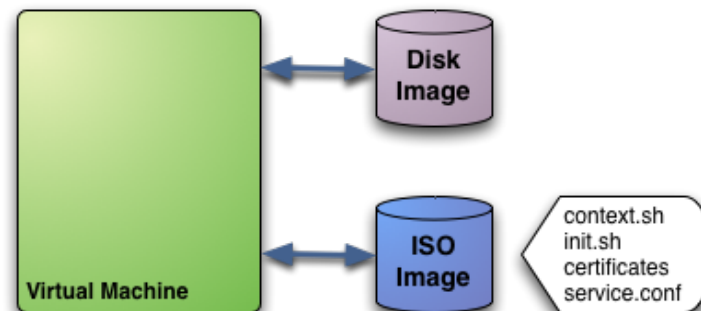
- Contextualizar una máquina virtual significa, en otras palabras, personalizarla a nuestra medida.
- Se trata ya de temas más avanzados.
- La imagen que hayamos creado tiene que estar preparada para interactuar con OpenNebula, debe tener instalado el paquete [opennebula-context](#).
- Más información en [http://docs.opennebula.org/4.12/user/virtual\\_machine\\_setup/cong.html](http://docs.opennebula.org/4.12/user/virtual_machine_setup/cong.html)

# Contextualizar

- Entre las cosas que podemos hacer con ello, destacamos las siguientes:
  - Cambiar la configuración de red de la máquina, así adaptarla a la red en la que se encuentre el host, en función de los parámetros de red que hayamos configurado y las redes que le hayamos configurado.
  - Introducir los parámetros deseados a la máquina virtual, como por ejemplo la clave SSH del nodo, de forma que no necesitemos contraseña a la hora de hacer SSH a la máquina virtual.
  - Introducir cualquier fichero deseado a la máquina. Esto da la opción de introducir cualquier script o archivo externo. Las posibilidades son infinitas.
  - Aprovechando la opción del script externo se puede usar una misma imagen para varias cosas, de forma que dependiendo del script inicial que ejecutemos la máquina virtual se prepare de una forma u otra, con el consiguiente ahorro de espacio. La alternativa a esto sería tener varias imágenes específicas, aunque el espacio consumido en el datastore aumentaría considerablemente.

# Contextualizar

- La contextualización se define a través de las plantillas. Esto creará una imagen de contextualización (que funciona como un CD para la máquina virtual, como se puede apreciar en la imagen) mediante el parámetro CONTEXT en las plantillas.
- Del parámetro CONTEXT colgarán diversos valores, como NETWORK, que indicará si se quiere o no contextualizar la red, SSH para la clave pública ssh o cualquier tipo de variable personalizada.
- Finalmente el valor que acompañe al atributo FILES serán los archivos (las rutas a ellos) que queremos pasar a las máquinas.
- Los scripts de los que hablábamos antes.



# Varios

- Destacar que existen muchas más órdenes además de las comentadas, ya sean de mantenimiento y control o de configuración de nuestro entorno.
- Por ejemplo podemos gestionar los datastores con el comando `onedatastore`, los nodos mediante `onehost`, usuarios con `oneuser`, etc.
- Otra forma de monitorización son los logs, que se encuentran en `/var/log/one/`.

# Manos a la obra!

- Ahora te toca poner en marcha las máquinas virtuales con OpenNebula
- Sigue el tutorial que encontrarás en el guión de prácticas:  
[https://github.com/manuparra/MasterDegreeCC\\_Practice#starting-with-opennebula](https://github.com/manuparra/MasterDegreeCC_Practice#starting-with-opennebula)
- Sigue todos los pasos y comprueba que todo se crea correctamente.
- Recuerda que a la hora de configurar cada una de las MVs tendrás que hacer uso de nociones “avanzadas” de sysop.

# OpenNebula

Introducción al uso de Cloud Computing