

# DBigCloud



Tu Blog de sistemas

## INSTALANDO MIRANTIS OPENSTACK

Como montar un laboratorio de Cloud Computing casero

SEGUNDA EDICION

Daniel Romero Sánchez

[daniel.romero@dbigcloud.com](mailto:daniel.romero@dbigcloud.com)

<http://www.dbigcloud.com>

## ÍNDICE DE CONTENIDOS

1. EL AUTOR .....	1
2. PRÓLOGO .....	2
3. INTRODUCCIÓN .....	3
4. REQUISITOS.....	4
4.1. REQUISITOS HARDWARE .....	4
4.2. REQUISITOS SOFTWARE .....	4
5. PREPARANDO EL ENTORNO .....	5
5.1. INSTALANDO VIRTUALBOX.....	5
5.2. DESCARGA DE MIRANTIR OPENSTACK .....	6
6. INSTALACIÓN DE MIRANTIS OPENSTACK .....	7
6.1. ACCEDIENDO A FUEL .....	11
7. COMPONENTES DE OPENSTACK .....	12
7.1. COMPONENTES DEL MASTER FUEL .....	12
7.2. ARQUITECTURA .....	13
7.3. ROLES .....	14
8. CREACION DE UN ENTORNO OPENSTACK .....	15
8.1. PREPARACION DEL ENTORNO .....	15
8.2. DESPLEGANDO OPENSTACK.....	19
9. ACCEDIENDO A HORIZON.....	23
9.1. CREAR UN PROYECTO.....	23
9.2. CREAR UN USUARIO .....	26

## 1. EL AUTOR

Mi nombre [Daniel Romero Sánchez](#) y soy administrador de sistemas con más de 10 años de experiencia en el sector de las telecomunicaciones.

Durante todos estos años he ido adquiriendo altos conocimientos en **Sistemas Operativos** (Solaris, RedHat/Suse/Debian y Windows), **Bases de datos relacionales** (MySQL, Oracle, Postgresql) Cluster (RedHat Cluster, Oracle RAC, Sun Cluster, vSphere HA Cluster), **Gestión de cabinas de Almacenamiento** (EMC, NetApp, IBM Storwize, HP EVA, HP MSA), **Gestión copias de seguridad** (Veritas Netbackup, Bacula, HP Data Protector y Oracle backups RMAN/Datapump/Exports), **Scripting** (Bash, Python, Perl, PHP), **Sistemas de monitorización** (LostBit Monitor, Nagios, Centreon), etc.

He utilizado como metodología de trabajo principal ITIL, ISO 20.000 e ISO 27.001.

Me encanta estar al día en mi profesión, soy una persona autodidacta a la que le gusta aprender todo aquello que me haga ser mejor profesional y mejor persona.

Desde hace un año **escribo en los blog** [www.hispabigda.es](http://www.hispabigda.es) y <http://dbigcloud.com> **blog relacionados con el BigData y el CloudComputing**, ya que me gusta compartir mis conocimientos con todos aquellos que le puedan ser útil.

En 2015 he liberado mi primera aplicación para VMware, **DBC-VMfinder**, un buscador que nos va a permitir encontrar de forma fácil máquinas virtuales por MAC, IP o su nombre de nuestro Datacenter.

Mi dirección de correo electrónico de contacto es: [daniel.romero@dbigcloud.com](mailto:daniel.romero@dbigcloud.com)

## 2. PRÓLOGO

Gracias a la gran aceptación de los artículos de **Cómo instalar Mirantis OpenStack**, en nuestro portal web <http://www.dbigcloud.com>, he decidido lanzar este documento para tener una guía de referencia de como montar un laboratorio de Cloud Computing casero totalmente en español.

El siguiente documento no es una guía oficial, ni pretende serlo, parte de los datos aportados han sido extraído de la guía oficial de Mirantis <http://docs.mirantis.com/openstack/fuel/fuel-5.1/> y está basado en la experiencia del equipo de [@DBigCloud](#).

Quiero agradecer la ayuda de **Rafa Hernández** [@rafah3ndez](#) y de **José M. Hernández** [@jos3mhb](#), ya que sin ellos esta guía no hubiese sido posible.

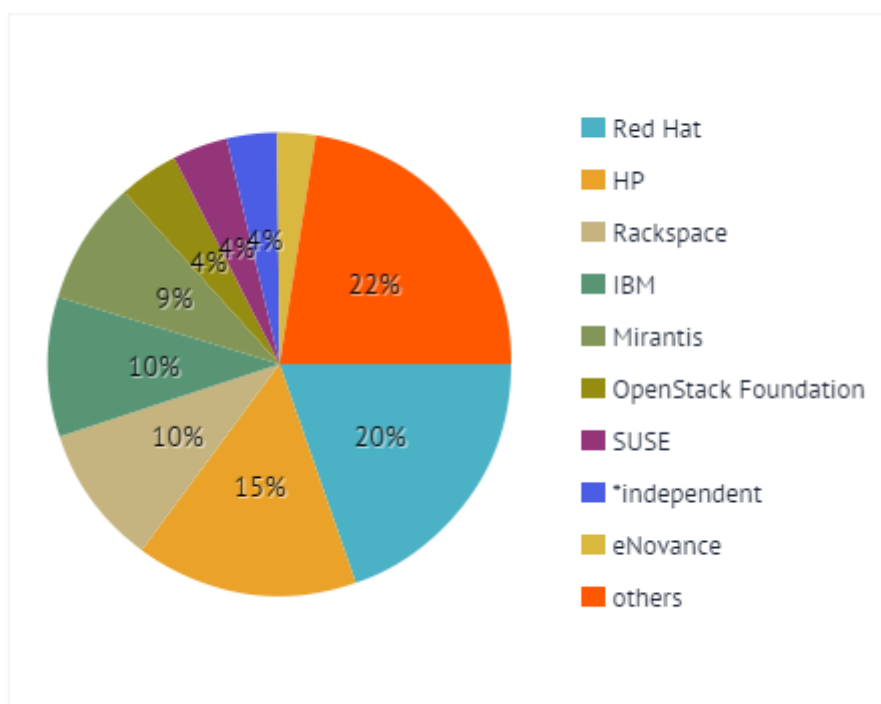
Si te gusta esta guía y puedes, puedes contribuir para mantener DBigCloud en marcha, con contenido nuevo semanalmente, a través de paypal en el portal web <http://www.dbigcloud.com>.

### 3. INTRODUCCIÓN

Hoy en día todo el mundo habla de la nube y de migrar los servicios a la nube, ¿Pero realmente sabemos que es lo que es? ¿Qué hay por debajo de la nube? A modo abreviado **la nube no es ni más ni menos que sistemas informáticos ofrecidos como servicio a través de internet**, sin entrar a valorar entre nube privada o nube pública.

Para ofrecer esto, existen muchas herramientas tanto de pago como gratuitas como son [vCloud](#), [Openstack](#) o [CloudStack](#) entre las más conocidas. Cada una de ellas tiene sus pros y sus contras, los cuales no vamos a analizar ya que no es el objetivo de esta guía. Pero si hay destacar que grandes empresas como RedHat, Mirantis, HP, Suse, etc. han apostado fuerte por OpenStack que es OpenSource, como podemos apreciar en la siguiente gráfica.

Contribution by companies



Nosotros os vamos a explicar **como realizar una instalación básica de OpenStack con la distribución Mirantis bajo la herramienta de desarrollo y administración Fuel** ya que es bastante sencilla de instalar y así de esta forma podéis montar vuestros laboratorios de iniciación de sistemas en la nube.

## 4. REQUISITOS

Montar un sistema de producción de OpenStack es bastante costoso ya que los recursos necesarios para ofrecer un servicio decente son bastante altos.

Nuestra intención es montar un laboratorio casero y para ello vamos a usar las posibilidades que nos ofrece la virtualización, por lo que los requisitos en los que nos vamos a basar únicamente serán recomendados para dicho entorno.

### 4.1. REQUISITOS HARDWARE

Como vamos a **virtualizar sobre virtualizado** (Nested virtualization) vamos a necesitar un equipo algo potente con al menos 8gb de RAM aunque lo recomendable serían 16gb, ya que tenemos que crear diferentes máquinas virtuales.

Las máquinas virtuales serán desplegadas de forma automática como veremos más adelante y los recursos se le asignarán en base a los recursos físicos que dispongamos, aquí os mostramos un ejemplo para una instalación en un equipo con 8Gb de RAM.

#### Master Fuel

- 1 CPU
- 1536 Mb RAM
- 1 gigabit network port
- 65GB disco duro

#### Nodos

- 1 CPU
- 1024 Mb RAM
- 1 gigabit network port
- 65GB disco duro

### 4.2. REQUISITOS SOFTWARE

La instalación de Mirantis OpenStack la podemos realizar en cualquier distribución Linux en la cual podamos instalar la última versión de VirtualBox.

El software requerido es el siguiente:

- Distribución Linux (Centos, Ubuntu, Debian...)
- Development Tools
- kernel-devel
- expect
- VirtualBox-4.3
- VirtualBox ExtensionPack
- Activación de Dynamic Kernel Module Support

## 5. PREPARANDO EL ENTORNO

La distribución que vamos a usar para instalar nuestro entorno va a ser Centos 6.5, por lo que los comandos que vamos a utilizar son basados en yum y en rpm, pero si queréis usar cualquier otra distribución no tendréis ningún problema adaptando los comandos.

### 5.1. INSTALANDO VIRTUALBOX

El requisito principal es que tener instalada **la última versión de Virtual Box y extensión pack**, por lo que vamos a proceder a realizarlo, para ello vamos a ir realizando lo siguiente:

- Configurar el repositorio

```
cd /etc/yum.repos.d
wget http://download.virtualbox.org/virtualbox/rpm/rhel/virtualbox.repo
```

Una vez tengamos el repositorio tenemos que asegurarnos que está habilitado, para ello lo editamos y comprobamos que este la línea esta enable=1, en caso de estar a 0 lo cambiamos por 1.

- Lo siguiente es instalar las herramientas de desarrollo y del kernel, junto con el paquete expect.

```
yum groupinstall "Development Tools"
yum install kernel-devel
yum install expect
```

- Ahora instalamos VirtualBox:

```
yum install VirtualBox-4.3
```

- Una cosa importante será activar el **DKMS** (Dynamic Kernel Module Support).

```
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
wget http://pkgs.repoforge.org/rpmpforge-release/rpmpforge-release-0.5.3-1.el6.rf.x86\_64.rpm
```

```
rpm -i rpmforge-release-0.5.3-1.el6.rf.*.rpm  
yum --enablerepo rpmforge install dkms
```

- Lo siguiente será añadir nuestro usuario al grupo de usuarios de VirtualBox:

```
usermod -a -G vboxusers username
```

- Por último tenemos que instalar el extensión pack para ello nos lo descargamos desde <https://www.virtualbox.org/wiki/Downloads> y ejecutamos lo siguiente:

```
VBoxManage extpack install /RUTA/Oracle_VM_VirtualBox_Extension_Pack-4.3.14-95030.vbox-  
extpack
```

## 5.2. DESCARGA DE MIRANTIR OPENSTACK

Ahora vamos a **descargar la distribución de Mirantis OpenStack 6.0 JUNO** que podemos realizarla desde <https://software.mirantis.com/>.



# GET MIRANTIS OPENSTACK

Choose format and content to download:

☒ .ISO

☐ .IMG

☐ Upgrade Package

GET 6.0 ISO

A choice of formats provides you with a **simple, convenient package** to start your OpenStack cloud using intuitive automated deployment by Fuel.

Or try this: [DOWNLOAD ISO FROM TORRENT](#)

Choose 'Upgrade Package' if you're upgrading from 5.x to 6.0.

QUICK START

The Quick Start Guide lets you get from **0 to OpenStack in 60 Minutes or less** with a typical laptop or small server. Choose between Virtual Box scripts or a vApp appliance.

READ THE DOC

Got bigger plans beyond a quick start? Read the documentation for complete details and reliable **Mirantis OpenStack Reference Architectures**.

Una vez ya tenemos descargada la distribución de OpenStack Mirantis, tenemos que descargarnos los scripts que nos van a ayudar a la instalación:

```
wget http://9f2b43d3ab92f886c3f0-  
e8d43ffad23ec549234584e5c62a6e24.r60.cf1.rackcdn.com/vbox-scripts-5.0.1.zip
```

## 6. INSTALACIÓN DE MIRANTIS OPENSTACK

Cuando descomprimos el fichero descargado anteriormente, se nos habrá creado una carpeta llamada `VirtualBox` que contendrá la siguiente estructura:

```
root@dbigcloud01:/opt/openstack/virtualbox# ls -ltra
total 48
-rw-r--r-- 1 root root 709 dic 25 15:27 README.md
-rwxr-xr-x 1 root root 1338 dic 25 15:27 launch.sh
-rwxr-xr-x 1 root root 43 dic 25 15:27 launch_8GB.sh
-rwxr-xr-x 1 root root 43 dic 25 15:27 launch_4GB.sh
-rwxr-xr-x 1 root root 44 dic 25 15:27 launch_16GB.sh
drwxr-xr-x 2 root root 4096 dic 25 15:27 iso
drwxr-xr-x 2 root root 4096 dic 25 15:27 functions
-rwxr-xr-x 1 root root 6087 dic 25 15:27 config.sh
drwxr-xr-x 2 root root 4096 dic 25 15:27 actions
drwxr-xr-x 5 root root 4096 dic 25 15:27 .
drwxr-xr-x 3 root root 4096 ene 9 17:27 ..
```

Ahora hay que mover la ISO descargada anteriormente y colocarla en la ruta:

*PATH/virtualbox/iso/*

En la imagen anterior se puede observar cómo queda la estructura completa.

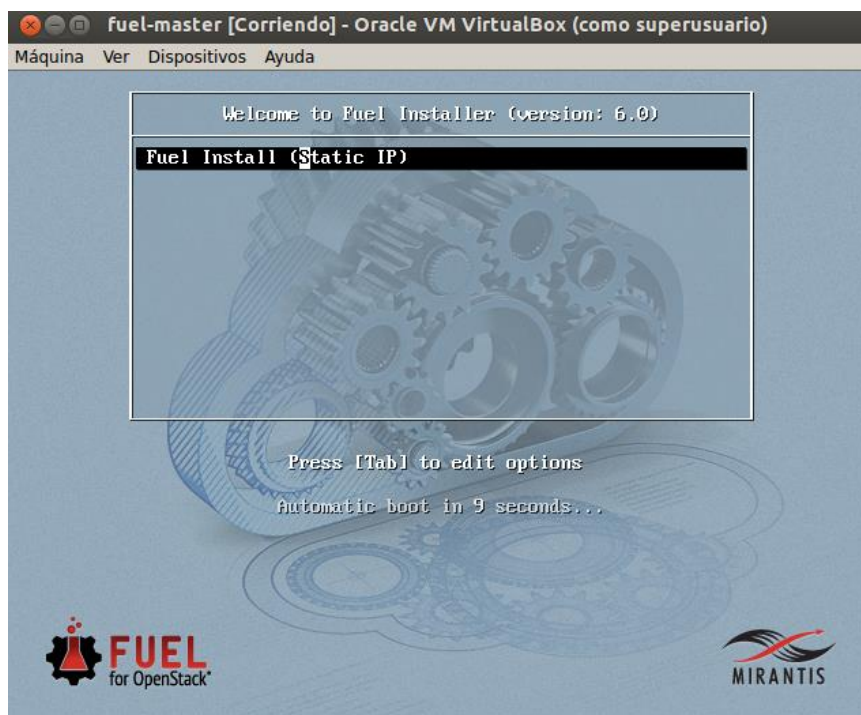
En la carpeta VirtualBox existen diferentes ficheros, los más importantes:

- **config.sh** en el cual podremos establecer las opciones de instalación de los nodos que compondrán nuestro sistema, la memoria, el espacio asignado... Nosotros vamos a dejarlo por defecto.
- **launch.sh** es el encargado de provisionar las máquinas virtuales para montar nuestro entorno. Dependiendo de la cantidad de memoria RAM que dispongamos en nuestro equipos podemos ejecutar cualquiera de los **launch\_xx.sh**.

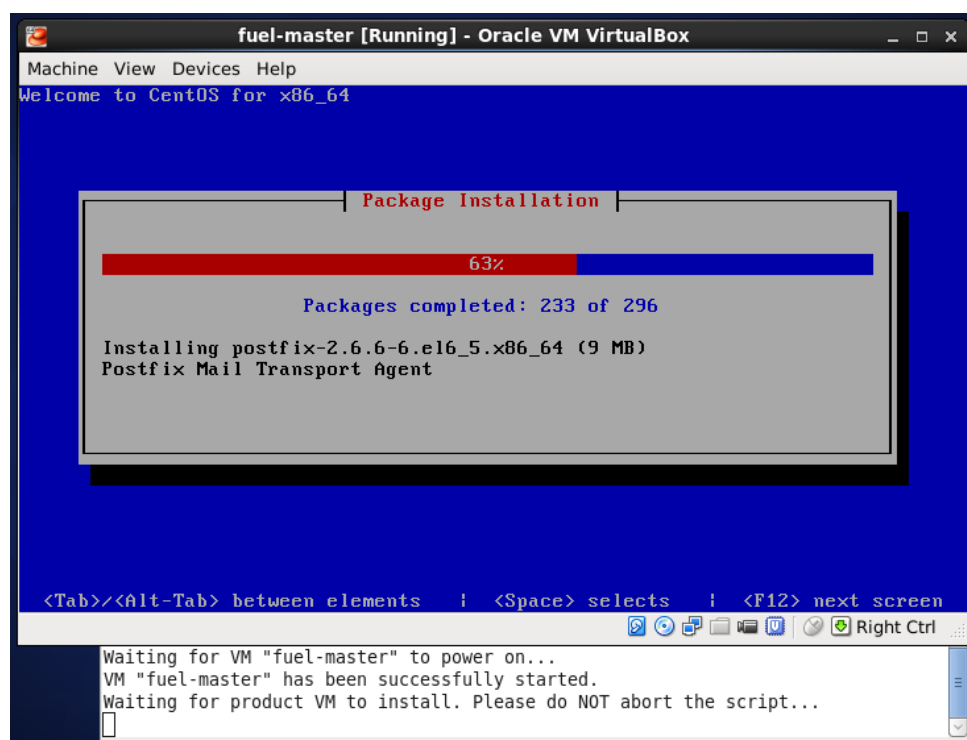
Para realizar la instalación ejecutamos lo siguiente:

```
./launch.sh
```

Se nos abrirá una ventana de VirtualBox para instalar Fuel.



Este proceso durará varias horas, dependiendo de los recursos del servidor donde se vaya a instalar, ya **que desplegará las máquinas virtuales que compondrán nuestra infraestructura de OpenStack, y se instalará el Nodo Master Fuel**, por lo que no debemos interrumpir el script en ningún momento, es recomendable que de vez en cuando vayáis viendo cómo va la instalación y que no hayáis recibido ningún error.



Una vez finalizado en la ventana de la máquina virtual del Nodo Master veremos la URL para acceder a FUEL y los datos de usuario y contraseña de acceso a la consola, dicha contraseña deberíais de cambiarla.

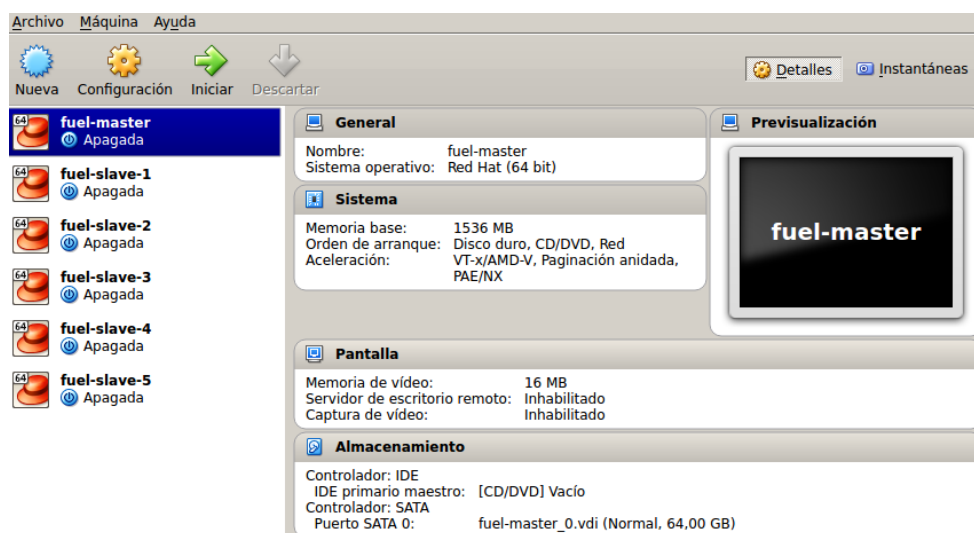
```
#####
#      Welcome to the Fuel server      #
#####
Server is running on x86_64 platform

Fuel UI is available on: http://10.20.0.2:8000

Default administrator login:   root
Default administrator password: r00tme

Please change root password on first login.
```

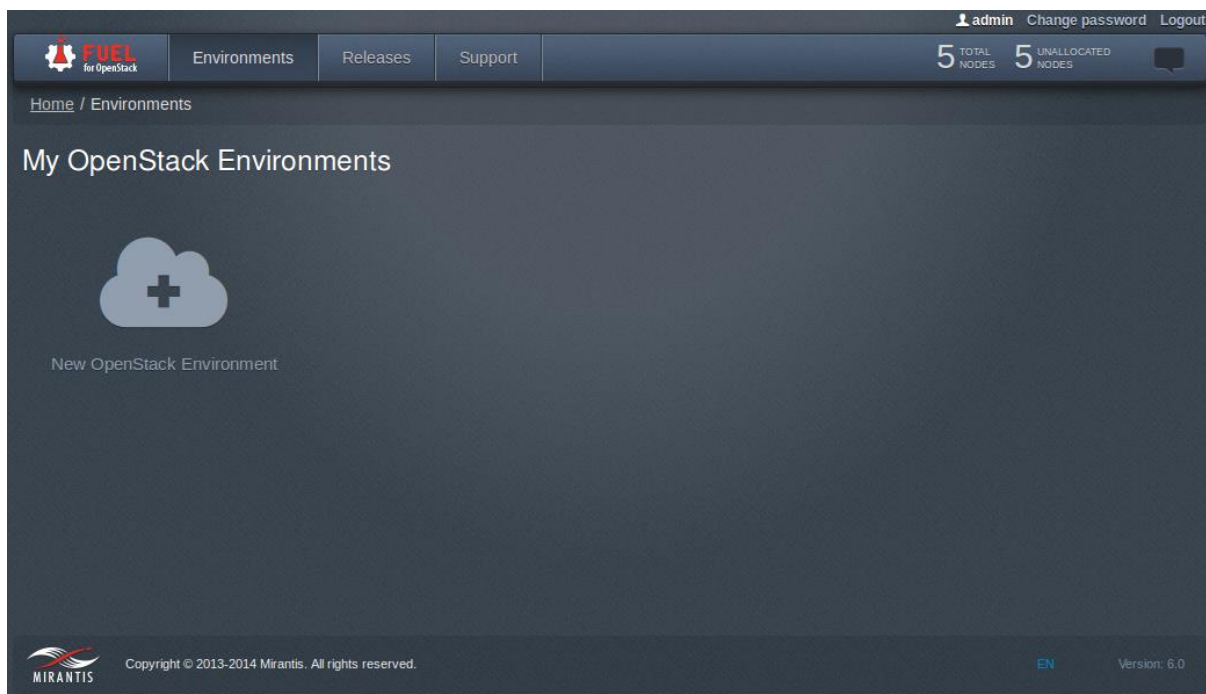
Ahora podemos comprobar en VirtualBox como se han creado todas las máquinas virtuales que formarán parte del entorno.



**Como podéis observar hay 6 máquinas virtuales, 1 Master y 5 nodos**, en la imagen podréis ver que se han apagado 2 de los nodos esclavos, ya que hasta la creación del entorno no nos va a hacer falta. Estos nodos se replican por PXE a través del nodo master.

## 6.1. ACCEDIENDO A FUEL

Ahora es el momento de **acceder a fuel** entrando en el navegador en la URL <http://10.20.0.2:8000>

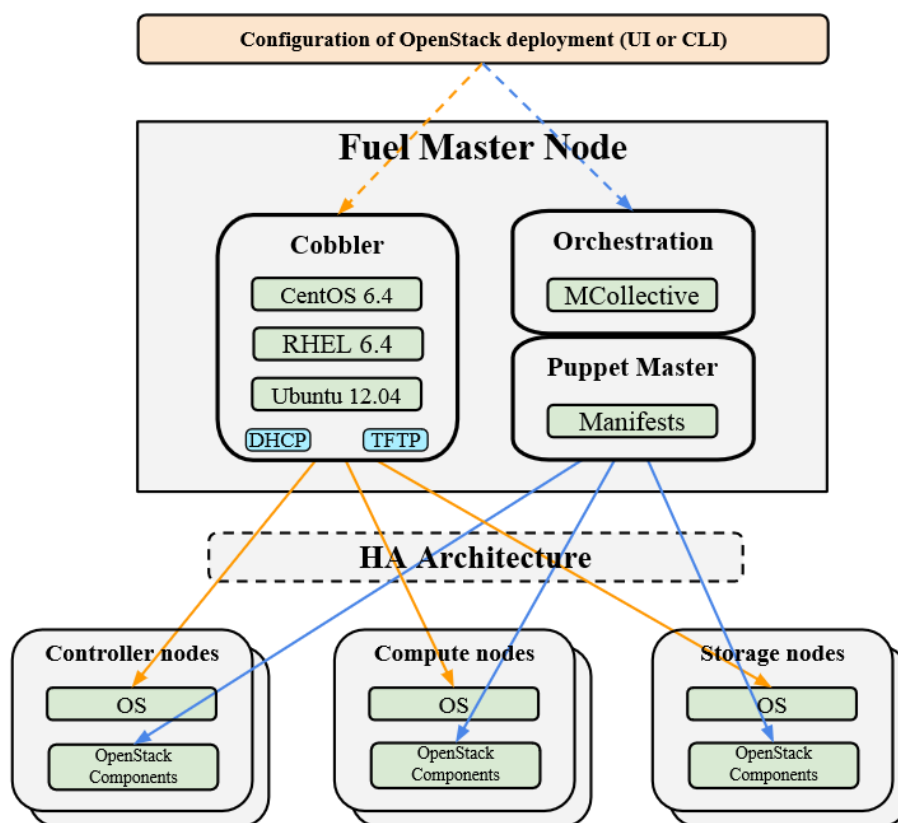


Como podéis ver en la parte superior derecha nos aparecerá una alerta y es para indicarnos que el nodo master se ha instalado correctamente, os irán apareciendo una notificación más por cada nodo que se vaya levantando, con esto ya disponemos de Fuel instalado en nuestro Nodo Master.

## 7. COMPONENTES DE OPENSTACK

Para entender el funcionamiento interno de **Fuel** y **OpenStack** nos vamos a basar en el esquema de la [documentación original de Mirantis](#).

Como podéis observar existen dos partes principales, que son Fuel como Nodo Master y los Nodos esclavos de OpenStack.



### 7.1. COMPONENTES DEL MASTER FUEL

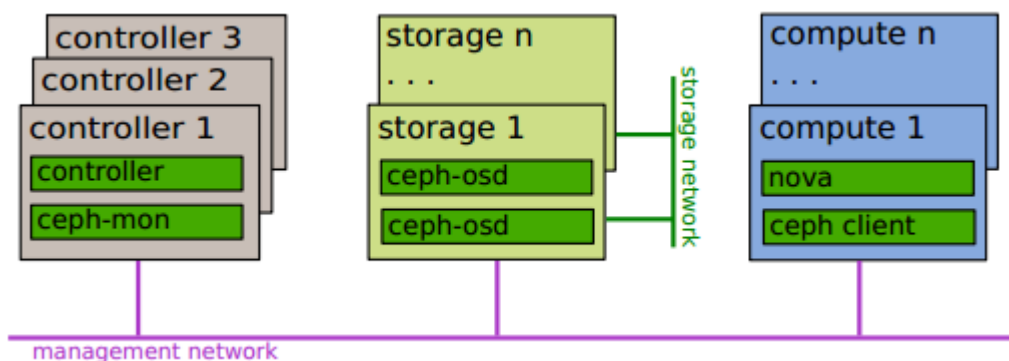
- **Cobbler** es el componente que se encargará de la provisión de los sistemas operativos de nuestros nodos, además de la designación del direccionamiento de IP.
- **Orchestration** es el componente que se encarga de comunicar todos los procesos, para realizar despliegues de servicios o recursos.
- Cuando el usuario a través de la interfaz gráfica decide desplegar algún servicio, se generan una serie de parámetros de configuración de recursos del sistema y estados que se almacenan en un fichero **Manifest**, para luego ser desplegado. El componente que se encarga de esto es **Puppet**.

## 7.2. ARQUITECTURA

Para entender en qué consisten los nodos esclavos, primero vamos a ver las opciones que nos ofrece la arquitectura de OpenStack.

- **Distribución linux** que instalaremos en nuestro entorno, pudiendo elegir entre Centos o Ubuntu, teniendo en cuenta que únicamente esta última trae soporte nativo para VLANs.
- La instalación de nuestro entorno la podemos realizar a través de una **arquitectura en alta disponibilidad**, teniendo diferentes nodos (que es lo recomendado en un sistema de producción) o bien en un solo nodo, dando a este toda la responsabilidad para los despliegues.
- Como **Hypervisor** de máquinas virtuales podremos elegir entre [KVM](#), [QMU](#) o [vCenter con ESXi](#).
- **Arquitectura de red** podemos elegir entre 4 topologías de red como son **Neutron** con [VLAN](#) o [GRE](#) segmentación, o **Nova-NetWork** con topología [VLAN](#) o [FlatDHCP](#), dependiendo de las necesidades que dispongamos nos decantaremos por una u otra.
  1. **OVS (Open vSwitch) y Bonding** solo puede ser usado con Neutron.
  2. **VMWare vCenter** solo se puede usar con Nova-network.
  3. **Murano** solo es soportado con Neutron.
- **Almacenamiento** que puede ser de dos tipos:
  1. [Cinder](#) que por defecto usa **LVM** para el almacenamiento por bloques compartidos por iSCSI o bien [Ceph](#) el cual es requerido para KVM.
  2. [Glance](#) que por defecto usa [Swift](#) por defecto o bien [Ceph](#).
- **Servicios Adicionales** que nos van a permitir usar servicios que no vienen por defecto en OpenStack, como son:
  1. [Sahara](#) nos permite provisionar bajo demanda clusters de hadoop.
  2. [Murano](#) provee un catálogo de aplicaciones que pueden ser desplegadas en la nube por los usuarios.
  3. [Ceilometer](#) provee un sistema de monitorización de nuestra nube OpenStack.

Una vez visto de forma general las opciones que disponemos para montar nuestra arquitectura nos será más sencillo determinar que roles instalaremos en cada nodo, teniendo en cuenta que para una arquitectura en **Alta Disponibilidad (HA)** debemos tener al menos 3 nodos con los siguientes roles:



### 7.3. ROLES

- **Controller o Controlador** es el que se encarga de la comunicación de las operaciones entre los servicios de la nube.
- **Storage o Almacenamiento** provee el almacenamiento al entorno ya sea por bloques o por objetos de almacenamiento.
- **Compute o Computador** es el encargado de proveer al hypervisor usado los recursos.

**Los roles de Almacenamiento pueden ser combinados con los Controladores o Computadores, dependiendo del tipo de almacenamiento elegido.** De esta forma podemos tener varios controladores redundados entre los nodos.

A través de la siguiente tabla nos puede guiar para determinar el número nodos de Controladores, Almacenamiento y Computadores que debemos tener.

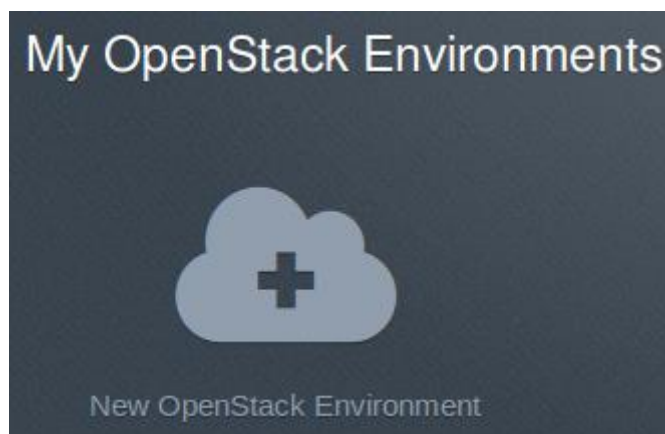
# of Nodes	Controllers	Computes	Storages
4-10	3	1-7	3 (on controllers)
11-40	3	3-32	3+ (swift) + 2 (proxy)
41-100	4	29-88	6+ (swift) + 2 (proxy)
>100	5	>84	9+ (swift) + 2 (proxy)



## 8. CREACION DE UN ENTORNO OPENSTACK

### 8.1. PREPARACION DEL ENTORNO

Para crear nuestro entorno debemos establecer la arquitectura que vamos a utilizar y que componentes vamos a usar, para ellos tenemos que dirigirnos al panel de administración de Fuel <http://10.20.0.2:8000> a través del navegador y nos dirigimos a *New OpenStack Environment*.



Y se nos abrirá una ventana en la que iremos eligiendo los componentes que vamos a utilizar en nuestro entorno, lo primero de ello será establecer **el nombre de nuestro entorno y la distribución** que vamos a elegir para instalar en nuestros nodos. Para nuestro laboratorio vamos a elegir Ubuntu ya que como vimos en el anterior artículo traía soporte nativo para trabajar con VLAN, pero podéis usar Centos si queréis.

### Create a new OpenStack environment

**Name and Release**

Deployment Mode

Compute

Networking Setup

Storage Backends

Additional Services

Finish

**Name**

dbigcloud

**OpenStack Release**

Juno on Ubuntu 12.04.4 (2014.2-6.0) (default)

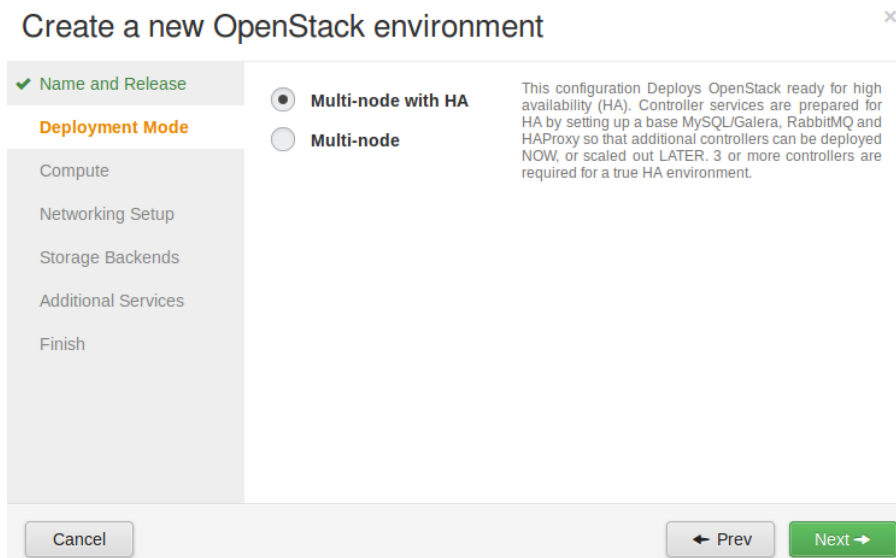
This option will install the OpenStack Juno packages using Ubuntu as a base operating system. With high availability features built in, you are getting a robust, enterprise-grade OpenStack deployment.

Cancel

← Prev

Next →

Una vez elegido el sistema operativo que se va a instalar en los nodos, tenemos que seleccionar si vamos a optar por un entorno en **HA (High Availability)**, o en multi nodo **sin HA**. Nosotros vamos a elegir la opción de **HA**, de esta forma tendremos alta disponibilidad.



Create a new OpenStack environment

✓ Name and Release

Deployment Mode

Multi-node with HA

Multi-node

This configuration Deploys OpenStack ready for high availability (HA). Controller services are prepared for HA by setting up a base MySQL/Galera, RabbitMQ and HAProxy so that additional controllers can be deployed NOW, or scaled out LATER. 3 or more controllers are required for a true HA environment.

Compute

Networking Setup

Storage Backends

Additional Services

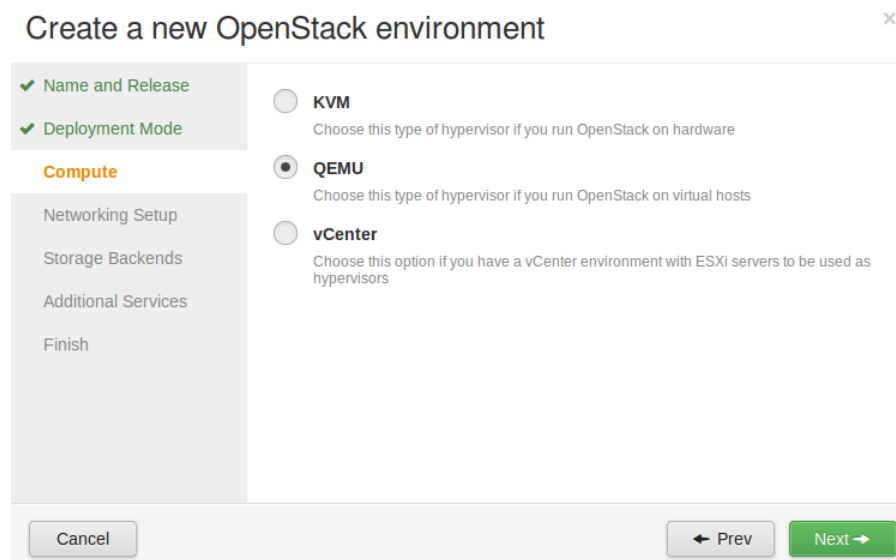
Finish

Cancel

Prev

Next

Ahora es el turno de escoger que sistema de virtualización vamos a usar, si tenéis un entorno con **KVM** o un **vCenter con ESXi** podéis elegir estas opciones, nosotros como nuestros nodos son virtualizados vamos a usar **QEMU**.



Create a new OpenStack environment

✓ Name and Release

✓ Deployment Mode

Compute

Networking Setup

Storage Backends

Additional Services

Finish

KVM

Choose this type of hypervisor if you run OpenStack on hardware

QEMU

Choose this type of hypervisor if you run OpenStack on virtual hosts

vCenter

Choose this option if you have a vCenter environment with ESXi servers to be used as hypervisors

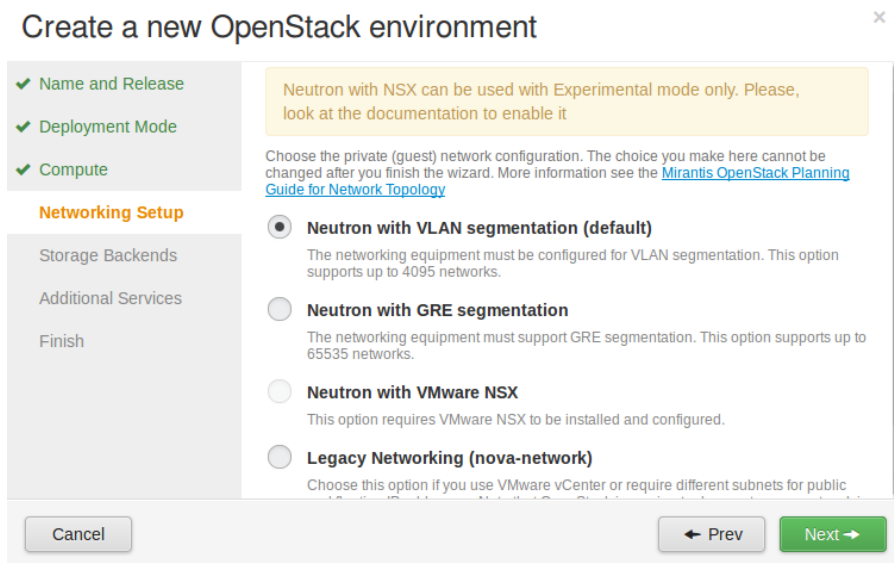
Cancel

Prev

Next

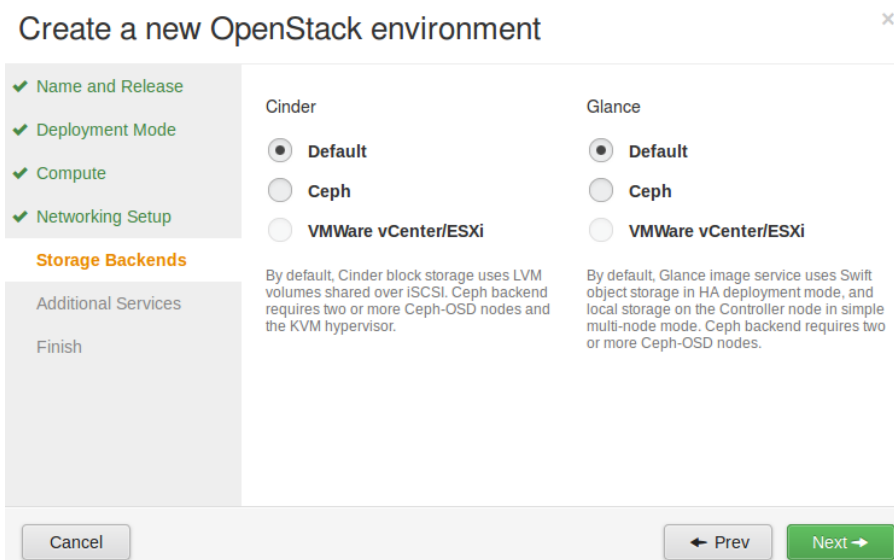
Llega el momento de decidir la arquitectura de red, aquí tenéis que tener en cuenta que si habéis elegido anteriormente la opción de **vCenter**, **debereis de elegir Nova-Network**, como nosotros vamos

a usar QEMU y posteriormente vamos a instalar **Murano**, elegiremos la opción de **Neutron with VLAN segmentation**.



The screenshot shows the 'Create a new OpenStack environment' wizard at the 'Networking Setup' step. The left sidebar shows the progress: 'Name and Release', 'Deployment Mode', 'Compute', 'Networking Setup' (active), 'Storage Backends', 'Additional Services', and 'Finish'. The main content area has a yellow warning box at the top: 'Neutron with NSX can be used with Experimental mode only. Please, look at the documentation to enable it'. Below this, a text block says: 'Choose the private (guest) network configuration. The choice you make here cannot be changed after you finish the wizard. More information see the [Mirantis OpenStack Planning Guide for Network Topology](#)'. There are four radio button options: 'Neutron with VLAN segmentation (default)' (selected), 'Neutron with GRE segmentation', 'Neutron with VMware NSX', and 'Legacy Networking (nova-network)'. Each option has a brief description. At the bottom, there are 'Cancel', 'Prev', and 'Next' buttons.

En cuanto a almacenamiento vamos a dejar las **opciones por defecto**, tanto para **Cinder** como para **Glance**.



The screenshot shows the 'Create a new OpenStack environment' wizard at the 'Storage Backends' step. The left sidebar shows the progress: 'Name and Release', 'Deployment Mode', 'Compute', 'Networking Setup', 'Storage Backends' (active), 'Additional Services', and 'Finish'. The main content area is divided into two columns: 'Cinder' and 'Glance'. Each column has three radio button options: 'Default' (selected), 'Ceph', and 'VMWare vCenter/ESXi'. Below the options, there is explanatory text for each service. At the bottom, there are 'Cancel', 'Prev', and 'Next' buttons.

Para **los servicios adicionales** tenemos la opción de seleccionar **Sahara, Murano y Ceilometer**, nosotros no vamos a instalar Sahara ya que como venimos comentando no disponemos de tantos recursos y no vamos a desplegar ningún cluster de Hadoop.

### Create a new OpenStack environment ×

✓ Name and Release

✓ Deployment Mode

✓ Compute

✓ Networking Setup

✓ Storage Backends

**Additional Services**

Finish

☐ **Install Sahara**  
Sahara enables on demand provisioning of Hadoop clusters to be deployed on OpenStack utilizing a variety of vendor distributions.

☒ **Install Murano**  
Murano is an application catalog, which allows application developers and cloud administrators to publish various cloud-ready applications in a browsable categorized catalog, which may be used by the cloud users (including the inexperienced ones) to pick-up the needed applications and services and composes the reliable environments out of them in a "push-the-button" manner.

☒ **Install Ceilometer (OpenStack Telemetry)**  
Ceilometer provides metering and monitoring of an OpenStack cloud.

Cancel

← Prev

Next →

Y ya con esto tenemos seleccionados todos los componentes que vamos instalar en nuestro entorno, por ultimo pulsamos en el botón de create.

### Create a new OpenStack environment ×

✓ Name and Release

✓ Deployment Mode

✓ Compute

✓ Networking Setup

✓ Storage Backends

✓ Additional Services

**Finish**

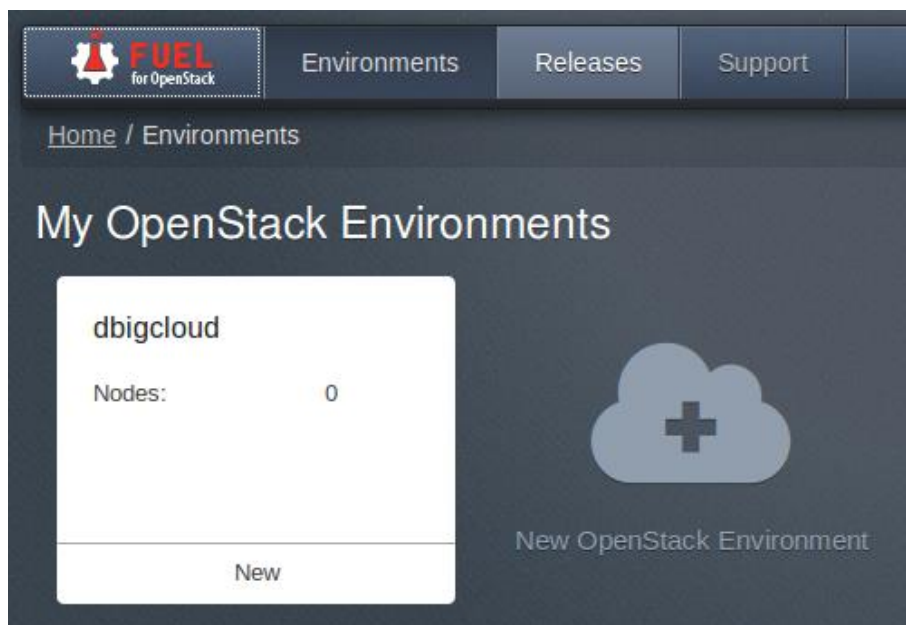
Your environment is now ready for deployment! After clicking on the Create button, you can select **Deploy Changes** or make additional configuration choices in the Fuel **Environments** console.

Cancel

← Prev

Create

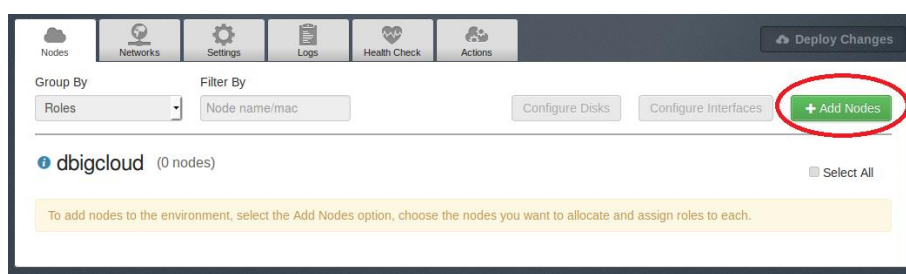
Con esto ya hemos creado el entorno, tal y como podéis observar en la siguiente imagen.



## 8.2. DESPLEGANDO OPENSTACK











Una vez tenemos creado el entorno, debemos acceder a él, para designar los componentes que se van a instalar en cada uno de los nodos, os tenéis que asegurar que estén encendidos, en caso de que no se encuentren encendidos os aparecerán mensajes indicándolo.

Una vez dentro, en la pestaña de Nodos podemos ir añadiendo cada uno de ellos, para ello pulsaremos sobre el botón de añadir nodos.



Y nos mostrará los tres nodos que creamos cuando realizamos la instalación del Master Fuel, cada uno de ellos, aparecerá sin roles asignados.

HDD: 192.0 GB RAM: 2.0 GB (5) ☐ Select All

<input type="checkbox"/>	 <b>dbc-ctropen01</b> UNALLOCATED	DISCOVERED	CPU: 0 (1) HDD: 192.0 GB RAM: 2.0 GB	
<input type="checkbox"/>	 <b>dbc-ctropen02</b> UNALLOCATED	DISCOVERED	CPU: 0 (1) HDD: 192.0 GB RAM: 2.0 GB	
<input type="checkbox"/>	 <b>dbc-ctropen03</b> UNALLOCATED	DISCOVERED	CPU: 0 (1) HDD: 192.0 GB RAM: 2.0 GB	
<input type="checkbox"/>	 <b>dbc-comopen01</b> UNALLOCATED	DISCOVERED	CPU: 0 (1) HDD: 192.0 GB RAM: 2.0 GB	
<input type="checkbox"/>	 <b>dbc-comopen01</b> UNALLOCATED	DISCOVERED	CPU: 0 (1) HDD: 192.0 GB RAM: 2.0 GB	

Seleccionaremos cada uno de ellos y le iremos asignado los roles, nosotros vamos a realizar la instalación de la siguiente forma:

- Asignamos a dos nodos el **ROL de Compute**.
- Uno de los nodos tendrá los **Roles de Controller, Storage - Cinder y Telemetry-MongoDB**
- A los otros dos le asignamos el **ROL de Controller, Storage – Cinder y Storage – Ceph OSD**.

Assign Roles

☐ **Controller**  
The controller initiates orchestration activities and provides an external API. Other components like Glance (image storage), Keystone (identity management), Horizon (OpenStack dashboard) and Nova-Scheduler are installed on the controller as well.

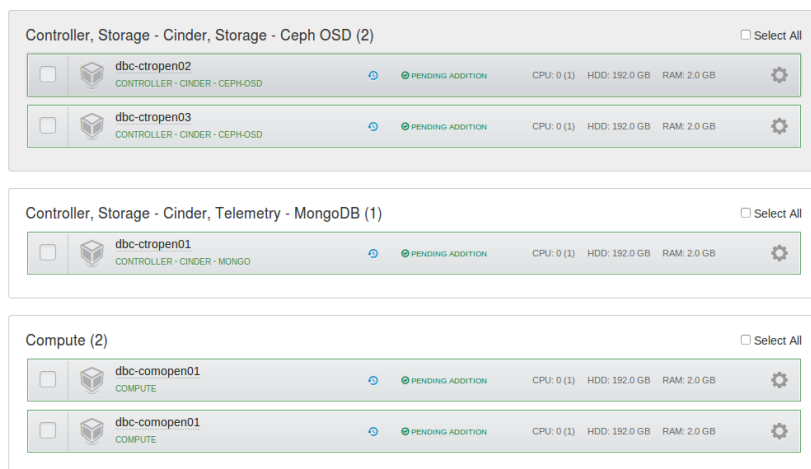
☐ **Compute**  
A compute node creates, manages and terminates virtual machine instances.

☐ **Storage - Cinder**  
Cinder provides scheduling of block storage resources, typically delivered over iSCSI, VMWare vCenter, and other compatible backend storage systems. Block storage can be used for database storage, expandable file systems, or providing a server with access to raw block level devices.

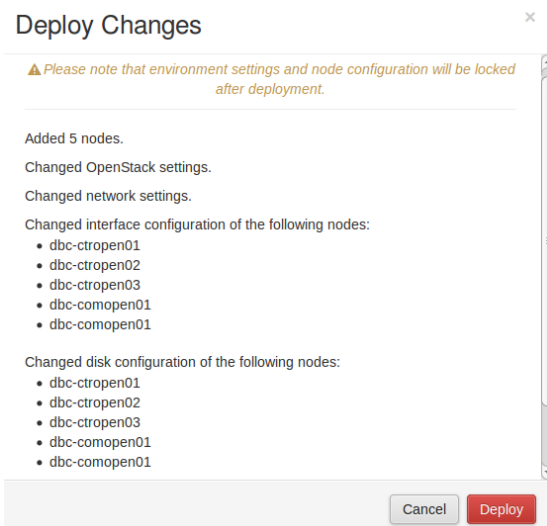
☐ **Storage - Ceph OSD**  
Ceph storage can be configured to provide storage for block volumes (Cinder), images (Glance) and ephemeral instance storage (Nova). It can also provide object storage through the S3 and Swift API (See settings to enable each).

☐ **Telemetry - MongoDB**  
A feature-complete and recommended database for storage of metering data from OpenStack Telemetry (Ceilometer).

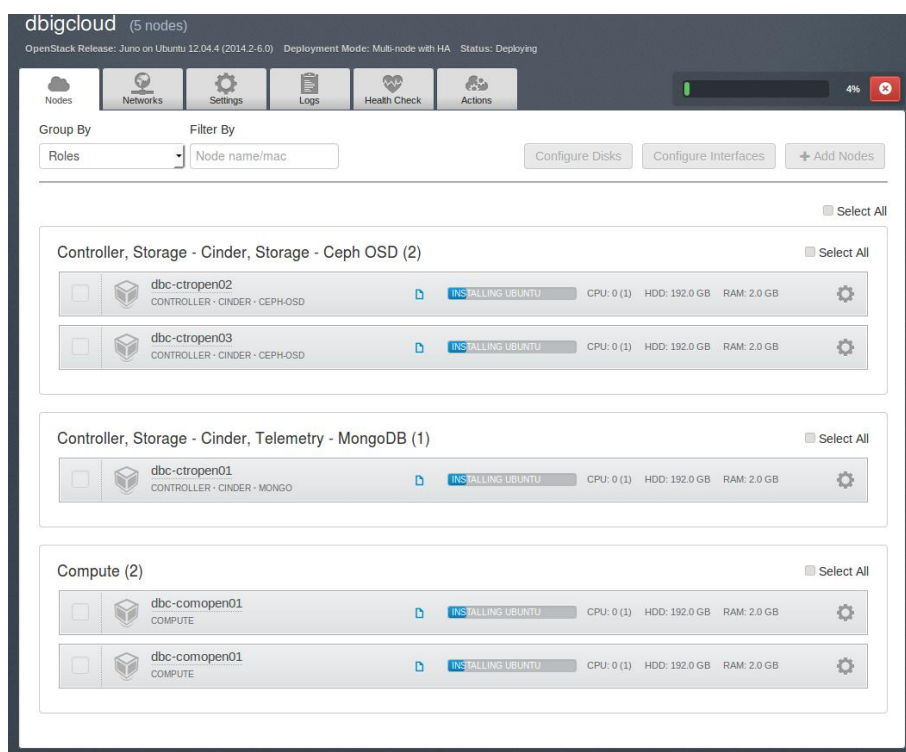
Una vez asignado los roles a los nodos nos quedará una cosa así:



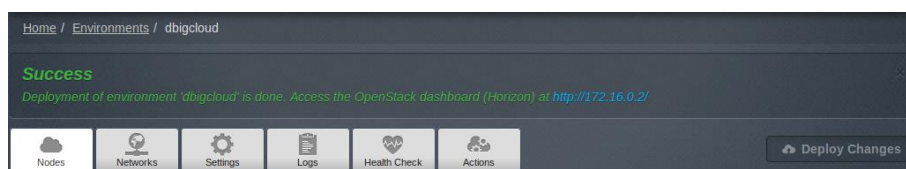
Por último pulsamos en la opción de **Deploy Changes**, nos mostrara una ventana a modo resumen de los nodos que se van a instalar, **una buena práctica sería renombrar cada uno de los nodos para identificarlos.**



La instalación de los nodos durará bastante tiempo, todo dependerá de las características de las máquinas virtuales que hemos desplegado anteriormente, el seguimiento de la instalación se puede ver a través de la pestaña de Log.



Una vez finalizada la instalación obtendremos un mensaje como el siguiente:

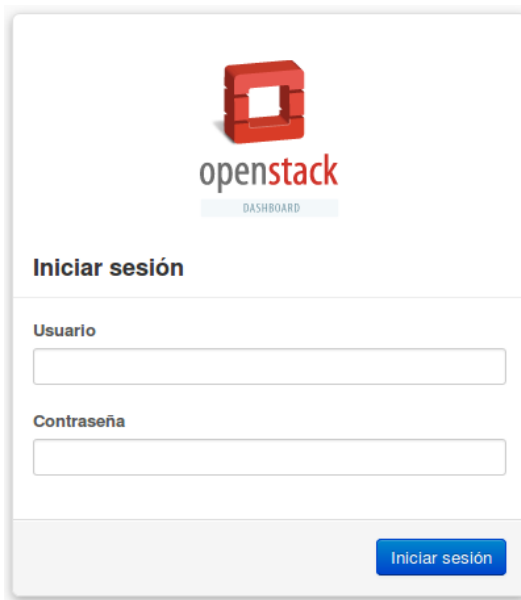


Ya por fin **tenemos instalado OpenStack** y podemos acceder, la siguiente buena práctica es dirigirse a la pestaña de setting y cambiar la password por defecto del usuario administrador de openstack.



## 9. ACCEDIENDO A HORIZON

Una vez desplegado todos los nodos con sus roles, podemos acceder a **Horizon**, la interfaz gráfica con la que podremos administrar nuestro entorno, si accedemos a través del navegador a la url indicada <http://172.16.0.2> ya podemos logarnos.



### 9.1. CREAR UN PROYECTO

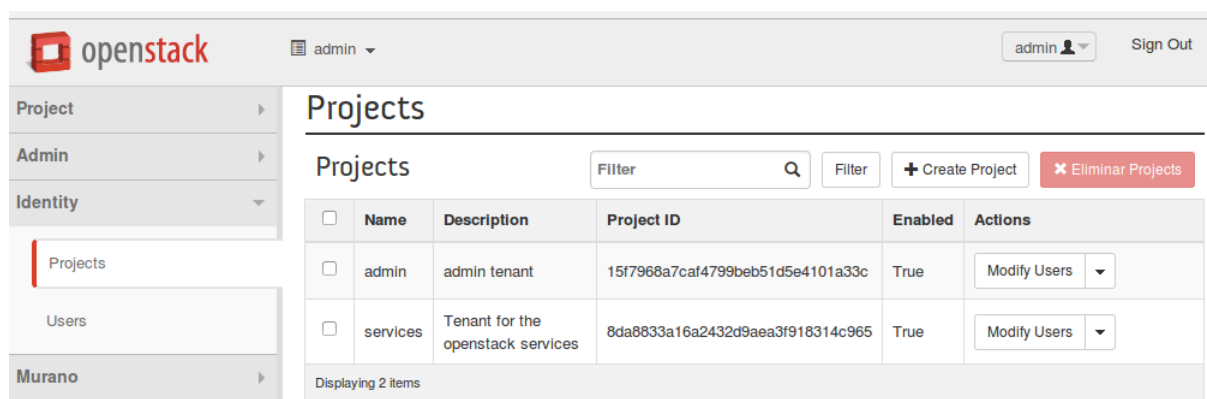
En OpenStack **los tenant, son llamados proyectos**, los cuales van a permitir a nuestros usuarios poder administrar su entorno.

A la hora de crear un proyecto debemos tener claro que características tendrá como por ejemplo el **número de volúmenes** que podemos crear, el **tamaño** de los mismos, la cantidad de **routers** o **redes** entre otras cosas.

De esta forma los usuarios del proyecto creado tendrán podrán usar esas característica cuando accedan a **Horizon**.

Para crear un proyecto nuevo debemos acceder a la interfaz de **Horizon** con el usuario **admin**, que es el creado por defecto en nuestra instalación.

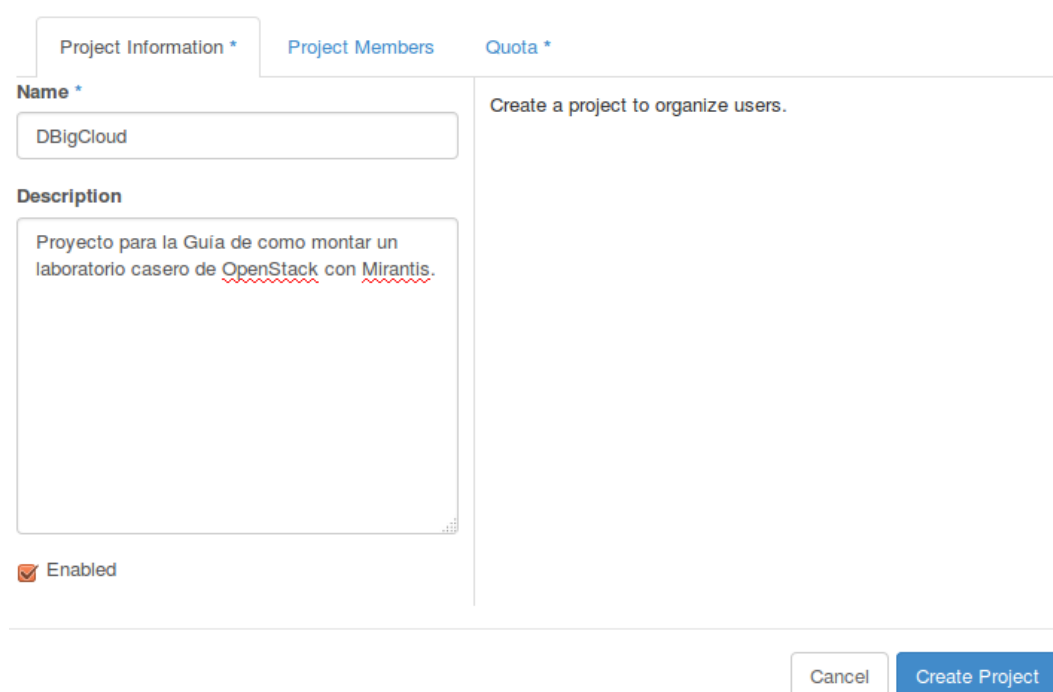
Una vez dentro de **Horizon**, debemos desplegar el menú de la izquierda en el elemento **Identity**, desde ahí podemos crear proyecto y usuarios.



	Name	Description	Project ID	Enabled	Actions
<input type="checkbox"/>	admin	admin tenant	15f7968a7caf4799beb51d5e4101a33c	True	Modify Users
<input type="checkbox"/>	services	Tenant for the openstack services	8da8833a16a2432d9aea3f918314c965	True	Modify Users

Si pulsamos sobre **Projects** se nos abrirá la ventana de administración de proyectos, si pulsamos sobre **Create Project**, se nos abrirá una nueva ventana en la que estableceremos las características de nuestro proyecto.

## Create Project



**Project Information** | Project Members | Quota

**Name \***  
DBigCloud

**Description**  
Proyecto para la Guía de como montar un laboratorio casero de OpenStack con Mirantis.

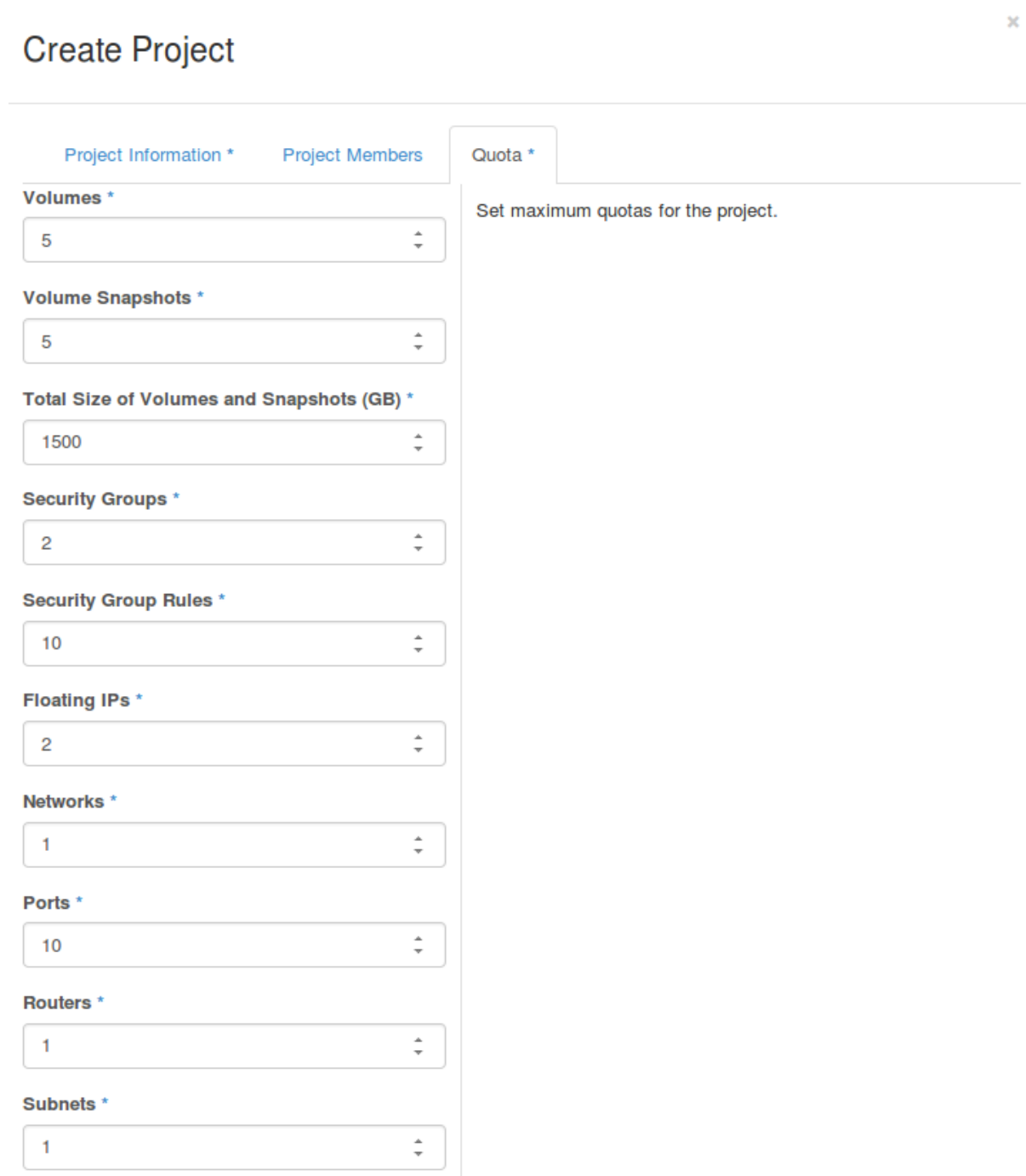
☒ Enabled

Cancel Create Project

En la pestaña **Project Information** rellenaremos el nombre del proyecto y su descripción, es importante establecer un nombre descriptivo y añadir una descripción en la que podamos identificar de forma rápida el proyecto.

La pestaña de **Project Memebtrs** la vamos a obviar por el momento, y nos dirigiremos a **Quota**, donde vamos a establecer las características del proyecto que estamos creando.

En la siguiente imagen podéis observar las opciones que disponemos:



## Create Project

Project Information \*

Project Members

Quota \*

**Volumes \***

5

**Volume Snapshots \***

5

**Total Size of Volumes and Snapshots (GB) \***

1500

**Security Groups \***

2

**Security Group Rules \***

10

**Floating IPs \***

2

**Networks \***

1

**Ports \***

10

**Routers \***

1

**Subnets \***

1

Set maximum quotas for the project.

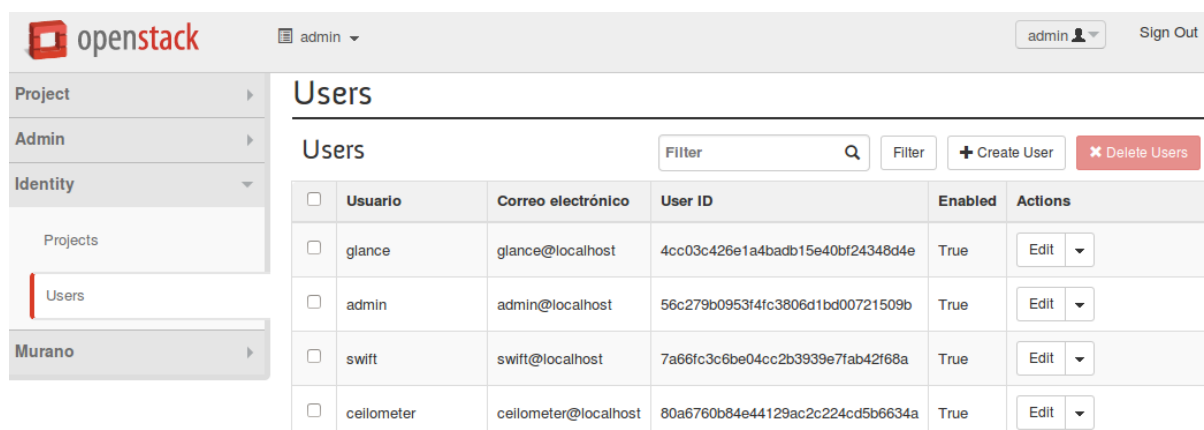
Por ejemplo he cambiado las **Quotas** por defecto limitándolas, como es un laboratorio casero no creo que necesitemos mucho más de los valores indicados, así de esta forma podemos tener diferentes proyectos sin sobrecargar mucho el entorno.

Una vez rellenado todo pulsamos en **Create Project** para que el proyecto **DBigCloud** sea creado.

## 9.2. CREAR UN USUARIO

Una vez creado el proyecto, debemos de crear un usuario para que pueda acceder, desde dentro del desplegable **Identity** debemos ir a **Users**.

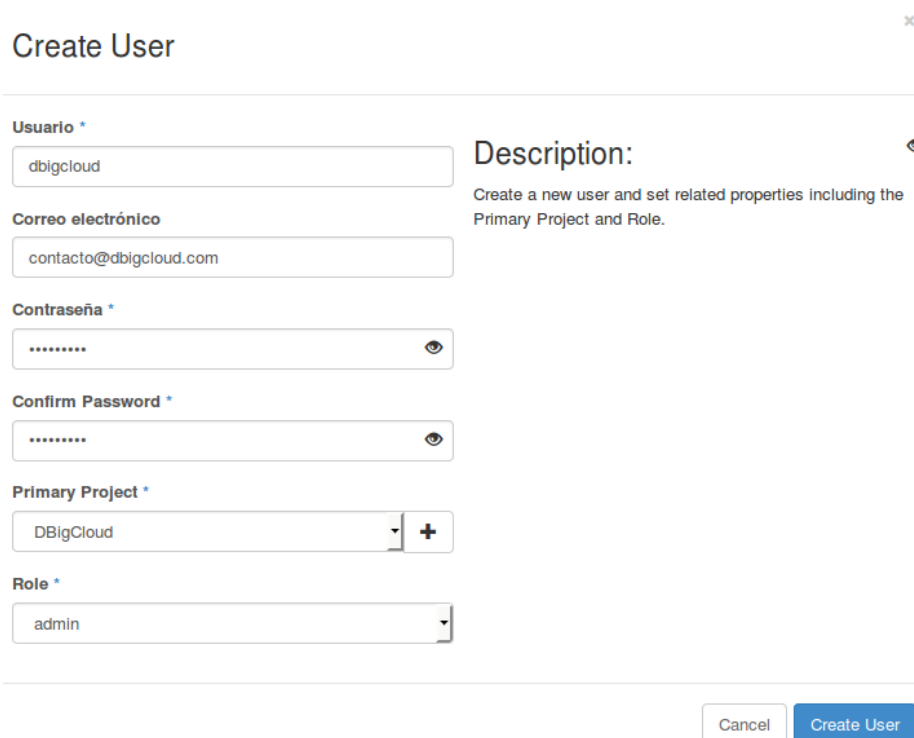
Se nos abrirá una ventana con el listado de todos nuestros usuarios, como veréis, hay distintos usuarios ya creados, los cuales son de los componentes de OpenStack.



The screenshot shows the OpenStack dashboard interface. On the left is a sidebar with navigation links: Project, Admin, Identity (selected), and Murano. The main content area is titled 'Users' and contains a table of existing users. Above the table are buttons for 'Filter', '+ Create User', and 'Delete Users'. The table has columns for 'Usuario', 'Correo electrónico', 'User ID', 'Enabled', and 'Actions'.

<input type="checkbox"/>	Usuario	Correo electrónico	User ID	Enabled	Actions
<input type="checkbox"/>	glance	glance@localhost	4cc03c426e1a4badb15e40bf24348d4e	True	Edit
<input type="checkbox"/>	admin	admin@localhost	56c279b0953f4fc3806d1bd00721509b	True	Edit
<input type="checkbox"/>	swift	swift@localhost	7a66fc3c6be04cc2b3939e7fab42f68a	True	Edit
<input type="checkbox"/>	ceilometer	ceilometer@localhost	80a6760b84e44129ac2c224cd5b6634a	True	Edit

Para crear uno nuevo debemos de pulsar en **Create User**, donde se nos abrirá una nueva ventana para rellenar los datos del nuevo usuario.



The 'Create User' form is displayed with a close button (X) in the top right corner. It contains several input fields and a description section.

**Usuario \***  
dbigcloud

**Correo electrónico**  
contacto@dbigcloud.com

**Contraseña \***  
.....

**Confirm Password \***  
.....

**Primary Project \***  
DBigCloud

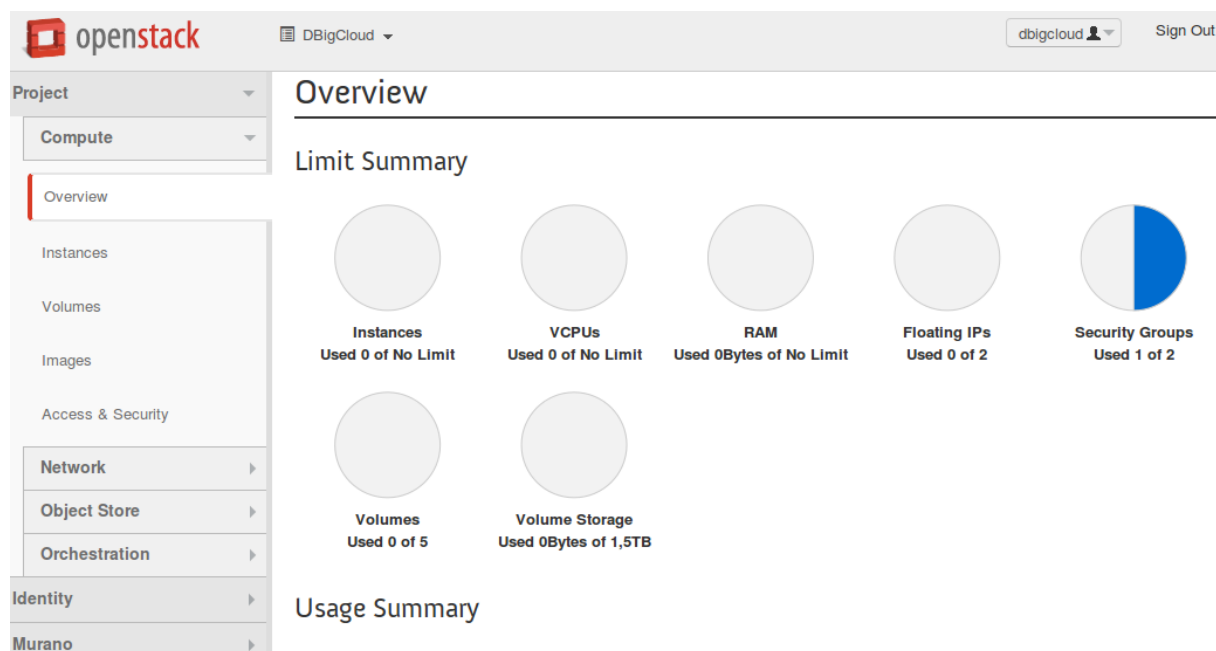
**Role \***  
admin

**Description:**  
Create a new user and set related properties including the Primary Project and Role.

Buttons: Cancel, Create User

En el campo **Primary Project** establecer el proyecto que hemos creado antes para **DBigCloud** y en el campo **Roles**, podemos indicar que permisos tendrá el usuario sobre el proyecto, en este caso vamos a crearlo de tipo **Admin**.

Una vez creado el usuario ya podemos acceder a nuestro tenant (proyecto) nos deslogamos con el usuario admin y accedemos con el usuario dbigcloud.



Con esto ya hemos creado nuestro proyecto, así de esta forma podemos practicar con OpenStack, que es el objetivo de esta guía.