

CLOUD COMPUTING: APACHE CLOUDSTACK



Elizabeth Moreno Begines

Índice:

<i>Introducción.....</i>	<i>3</i>
<i>Instalación y Configuración.....</i>	<i>4</i>
<i>Configuración de red.....</i>	<i>4</i>
<i>Configurando Hostname.....</i>	<i>6</i>
<i>Configurando SELinux.....</i>	<i>6</i>
<i>Configurando NTP.....</i>	<i>7</i>
<i>Instalando Apache CloudStack.....</i>	<i>7</i>
<i>Configuración BBDD.....</i>	<i>8</i>
<i>Configurando Servidor NFS.....</i>	<i>11</i>
<i>Configuración de plantilla de sistema.....</i>	<i>15</i>
<i>Instalando Hipervisor KVM.....</i>	<i>16</i>
<i>Configurando Apache CloudStack.....</i>	<i>17</i>
<i>Explorando Apache CloudStack.....</i>	<i>24</i>
<i>Creación de Máquinas virtuales.....</i>	<i>24</i>
<i>Creación de Espacios de Almacenamiento.....</i>	<i>26</i>
<i>Subida de Imágenes.....</i>	<i>29</i>
<i>Gestión de Usuarios.....</i>	<i>30</i>
<i>Creación de recursos para instancias.....</i>	<i>32</i>
<i>Bibliografía.....</i>	<i>35</i>

Introducción

Apache CloudStack es la plataforma líder de orquestación en la nube de código abierto, utilizada por muchas de las nubes públicas y privadas más grandes del mundo. El proyecto se creó después de la donación de Citrix del proyecto CloudStack a The Apache Software Foundation .

CloudStack es una plataforma de administración de la nube de Infraestructura como Servicio multi-hipervisor, multi-tenant, de alta disponibilidad. CloudStack es un software que proporciona una capa de orquestación en la nube, dando automatización en la creación, el aprovisionamiento y la configuración de componentes IaaS (infraestructura como servicio). Cloudstack convierte una infraestructura virtual existente en una plataforma basada en la nube como servicio (IaaS). El hecho de que CloudStack aproveche la infraestructura existente significa que el costo y el tiempo para que una organización cree una plataforma IaaS multi-tenant se reduce enormemente.

La arquitectura multi-tenant (multi-propietario) es aquella arquitectura donde todos los clientes y sus usuarios comparten todos los componentes tecnológicos que incluyen la aplicación, el modelo de datos, servidores y las capas de base de datos, el sistema de almacenamiento, entre otros. Además tienen la posibilidad de personalizar algunas partes de la aplicación, como el color de la interfaz de usuario (UI) o reglas de negocio, pero no pueden modificar el código de la aplicación.

Entonces, en lugar de que su equipo de ingeniería técnica tenga que configurar manualmente los servidores para los clientes, se pueden crear automáticamente en función de una compra en línea del cliente o una solicitud interna. Esto permite a los proveedores de servicios de TI y los equipos de infraestructura de TI de la empresa ofrecer verdaderas soluciones IaaS.

CloudStack es independiente del hipervisor y es compatible con VMware, Oracle VM, KVM, XenServer y Xen Cloud Platform. Esto le da a CloudStack una posición única en el mercado: no es tecnología patentada y no está vinculada a ninguna infraestructura específica.

Instalación y Configuración

La instalación comienza con el nodo de gestión "Management Server", para posteriormente detallar la configuración del hipervisor donde se ejecutarán las máquinas virtuales y finalmente, definir la configuración de Apache Cloudstack.

El servidor de gestión es el encargado de gestionar todas las operaciones que se van a realizar en la creación y configuración de la nube. Como servidor de gestión, se va a utilizar para el despliegue un Centos 6.9 de 64-bit. Es recomendable la asignación de al menos 4 GB de memoria RAM, 250 GB de espacio en disco y como mínimo, una interfaz de red estática. Estos valores son los mínimos necesarios para la creación de un entorno pequeño. Por el contrario, cuantos más recursos disponga el servidor, más rendimiento y escalabilidad tendrá el entorno.

Tras la instalación del sistema operativo, hay que realizar una serie de pasos y efectuar determinadas configuraciones para una correcta instalación de Apache CloudStack. Los pasos a seguir en la misma son los siguientes:

Iniciamos sesión en el servidor con el usuario root e instalamos un visor de documentos muy común (vim) con el fin de facilitar la realización de los pasos posteriores.

Antes de nada, instalaremos vim para poder editar los ficheros de configuración necesarios para el despliegue de cloudstack;

```
[root@cloudstack ~]# yum install vim
```

Configuración de red

Para la configuración de la red definiremos una IP estática a nuestro servidor de administración Apache Cloudstack. Para ello, editamos el fichero de configuración de la interfaz de red:

```
[root@cloudstack ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0
HWADDR=00:1E:68:B0:B9:4E
TYPE=Ethernet

NM_CONTROLLED=no

ONBOOT=yes
BRIDGE=cloudbr0
```

Configuramos, también, la interfaz que utilizaremos como bridge. Para ello, editamos el fichero de configuración de la interfaz de red:

```
[root@cloudstack ~]# vim /etc/sysconfig/network-scripts/ifcfg-cloudbr0
DEVICE=cloudbr0
HWADDR=00:1E:68:B0:B9:4E

TYPE=Bridge

NM_CONTROLLED=no

ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.1.200
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=8.8.8.8
DNS2=8.8.4.4
```

Una vez configuradas las interfaces, reiniciamos el servicio de red para aplicar los cambios:

```
[root@cloudstack ~]# chkconfig network on  
[root@cloudstack ~]# service network start  
Activación de la interfaz de loopback  
Activando interfaz eth0:  
  
Activando interfaz cloudbri0: Determining if ip address 192.168.1.200 is already in use for device cloudbri0...
```

RTNETLINK answers: File exists
RTNETLINK answers: File exists

Configurando Hostname

CloudStack requiere que el nombre del host esté configurado correctamente. Para comprobar el nombre del host de la máquina ejecutamos el siguiente comando:

```
[root@cloudstack ~]# hostname -fqdn  
hostname: `Host' desconocido
```

Procedemos a configurar el nombre del host. Primero editamos el fichero de configuración /etc/hostname y añadimos el nombre elegido:

```
[root@cloudstack ~]# vim /etc/hostname  
cloudstack
```

A continuación, editamos el fichero /etc/hosts y añadimos la IP de la máquina junto con el nombre definido anteriormente:

```
[root@cloudstack ~]# vim /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1      localhost localhost.localdomain localhost6 localhost6.localdomain6  
192.168.1.200 cloudstack
```

Por último, reiniciamos el servicio y comprobamos que se ha cambiado correctamente:

```
[root@cloudstack ~]# service network restart  
[root@cloudstack ~]# hostname --fqdn  
cloudstack
```

Configurando SELinux

Para que CloudStack funcione correctamente, SELinux debe estar configurado como permisivo.

Security-Enhanced Linux (SELinux) es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso.

Para que SELinux sea permisivo necesitamos ejecutar el siguiente comando:

```
[root@cloudstack ~]# setenforce 0
```

Para garantizar que permanezca en este estado permisivo, debemos configurar el fichero /etc/selinux/config:

```
[root@cloudstack ~]# vim /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=permissive
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Configurando NTP

La configuración NTP es necesaria para mantener sincronizados todos los relojes de los servidores en la nube. Sin embargo, NTP no está instalado por defecto. La instalación se realiza de la siguiente manera:

```
[root@cloudstack ~]# yum -y install ntp
```

La configuración predeterminada está bien para nuestros propósitos, por lo que simplemente necesitamos habilitarla y configurarla para que se inicie al arrancar.

```
[root@cloudstack ~]# chkconfig ntpd on
[root@cloudstack ~]# service ntpd start
    Iniciando ntpd:                                [ OK ]
```

Instalando Apache CloudStack

Tras estas configuraciones previas, es cuando procedemos a instalar el software de Apache CloudStack. Para ello, empezaremos por crear el repositorio de Cloudstack, añadiendo la siguiente ruta:

```
[root@cloudstack ~]# vim /etc/yum.repos.d/cloudstack.repo
[cloudstack]
name=cloudstack
baseurl=http://cloudstack.appt-get.eu/centos/6/4.6/
enabled=1
gpgcheck=0
```

Ahora procedemos a la instalación del nodo de gestión.

```
[root@cloudstack elizabeth]# yum install cloudstack-management
```

Configuración BBDD

El servidor de administración de CloudStack usa un servidor de base de datos MySQL para almacenar sus datos. Cuando instala el servidor de administración en un solo nodo, puede instalar el servidor de bbdd localmente.

De modo que, instalaremos el servidor de base de datos MySQL:

```
[root@cloudstack ~]# yum -y install mysql-server
```

Configuraremos los siguientes parámetros en el fichero de configuración de Mysql:

- *max_connections* debe establecerse en 350 multiplicado por la cantidad de Servidores de administración que está implementando. Este ejemplo supone un servidor de administración.
- *innodb_rollback_on_timeout* establece que el servidor se inicie con la opción de rollback.
- *innodb_lock_wait_timeout* indica el tiempo que va a mantenerse en espera hasta que un registro pueda quedar bloqueado.
- *log-bin* es la opción en que mysql guarda las transacciones de bbdd en los ficheros logs.
- *binlog-format* se utiliza para establecer el formato de registro.

```
[root@cloudstack ~]# vim /etc/my.cnf
[mysqld]
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=350
log-bin=mysql-bin
binlog-format = 'ROW'
```

Tras salir del fichero guardando la nueva configuración procederemos a reiniciar y habilitar el servicio en MySQL de la siguiente forma:

```
[root@cloudstack ~]# service mysqld restart
Iniciando base de datos MySQL: WARNING: The host 'cloudstack' could not be looked up
with resolveip.
This probably means that your libc libraries are not 100 % compatible
with this binary MySQL version. The MySQL daemon, mysqld, should work
normally with the exception that host name resolving will not work.
This means that you should use IP addresses instead of hostnames
when specifying MySQL privileges !
Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system
```

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !

To do so, start the server, then issue the following commands:

```
/usr/bin/mysqladmin -u root password 'new-password'  
/usr/bin/mysqladmin -u root -h cloudstack password 'new-password'
```

Alternatively you can run:

```
/usr/bin/mysql_secure_installation
```

which will also give you the option of removing the test databases and anonymous user created by default. This is strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:

```
cd /usr ; /usr/bin/mysqld_safe &
```

You can test the MySQL daemon with mysql-test-run.pl

```
cd /usr/mysql-test ; perl mysql-test-run.pl
```

Please report any problems with the /usr/bin/mysqlbug script!

[OK]

Iniciando mysqld:

[OK]

```
[root@cloudstack ~]# systemctl enable mysqld
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to  
/usr/lib/systemd/system/mariadb.service.
```

Una vez instalado el software de CloudStack, continuamos con la creación de la base de datos utilizada por CloudStack para almacenar los parámetros del entorno. Introduciremos la contraseña para acceder a la BBDD y el usuario que va a tener acceso a ella. Existen parámetros opcionales que nos permiten encriptar las contraseñas indicadas. Si se realiza este paso opcional, hay que indicar:

- *Tipo de encriptado. Este valor puede ser un fichero en una ruta conocida en cuyo caso habrá que poner el valor file e indicarle la correspondiente clave. La otra opción sería poner el valor web con lo que la herramienta utilizaría la función `com.cloud.utils.crypt.EncryptionSecretKeySender` para introducir la clave al servidor a través de un puerto en concreto. Por defecto, el valor es file.*

- *Clave del servidor de gestión. En este parámetro se indica la clave para encriptar la contraseña del servidor de gestión. Su valor por defecto es password.*
- *Clave de la base de datos. En este tercer parámetro se indica la clave para encriptar la contraseña de la base de datos. Su valor por defecto es password.*

Una vez analizados los diferentes parámetros, el comando final que se ejecuta es el siguiente sustituyendo los valores por los elegidos para el despliegue:

```
[root@cloudstack ~]# cloudstack-setup-databases cloud:cloudstack --deploy-as=root:
```

```
Mysql user name:cloud [ OK ]
Mysql user password:***** [ OK ]
Mysql server ip:localhost [ OK ]
Mysql server port:3306 [ OK ]
Mysql root user name:root [ OK ]
Mysql root user password:***** [ OK ]
Checking Cloud database files ... [ OK ]
Checking local machine hostname ... [ OK ]
Checking SELinux setup ... [ OK ]
Detected local IP address as 192.168.1.200,will use as cluster management server node IP [ OK ]
Preparing /etc/cloudstack/management/db.properties [ OK ]
Applying /usr/share/cloudstack-management/setup/create-database.sql [ OK ]
Applying /usr/share/cloudstack-management/setup/create-schema.sql [ OK ]
Applying /usr/share/cloudstack-management/setup/create-database-premium.sql [ OK ]
Applying /usr/share/cloudstack-management/setup/create-schema-premium.sql [ OK ]
Applying /usr/share/cloudstack-management/setup/server-setup.sql [ OK ]
Applying /usr/share/cloudstack-management/setup/templates.sql [ OK ]
Processing encryption ... [ OK ]
Finalizing setup ... [ OK ]
```

CloudStack has successfully initialized database, you can check your database configuration in /etc/cloudstack/management/db.properties

Ahora que la base de datos está configurada, terminamos de configurar el sistema operativo para el servidor de administración. Este comando configurará firewall, sudoers e iniciará el servidor de administración.

```
[root@cloudstack ~]# cloudstack-setup-management
Starting to configure CloudStack Management Server:
Configure Firewall ... [OK]
Configure CloudStack Management Server ...[OK]
```

CloudStack Management Server setup is Done!

Paramos los servicios de Cloudstack y lo volvemos a iniciar:

```
[root@cloudstack ~]# systemctl stop cloudstack-management.service
[root@cloudstack ~]# systemctl start cloudstack-management.service
```

Comprobamos que los servicios están funcionando correctamente:

```
[root@cloudstack ~]# systemctl status cloudstack-management.service
● cloudstack-management.service - CloudStack Management Server
   Loaded: loaded (/usr/lib/systemd/system/cloudstack-management.service; enabled;
  vendor preset: disabled)
   Active: active (running) since mié 2018-05-16 20:43:15 CEST; 9s ago
     Process: 2380 ExecStop=/usr/libexec/tomcat/server stop (code=exited,
status=0/SUCCESS)
   Main PID: 2514 (java)
    CGroup: /system.slice/cloudstack-management.service
            └─2514 /usr/lib/jvm/jre/bin/java -Djava.awt.headless=true
-Dcom.sun.management.jmxremote=false -Xmx2g -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/va...

may 16 20:43:20 cloudstack server[2514]: may 16, 2018 8:43:20 PM
org.apache.catalina.core.StandardEngine startInternal
may 16 20:43:20 cloudstack server[2514]: INFORMACIÓN: Starting Servlet Engine:
Apache Tomcat/7.0.76
may 16 20:43:20 cloudstack server[2514]: may 16, 2018 8:43:20 PM
org.apache.catalina.startup.HostConfig deployDirectory
may 16 20:43:20 cloudstack server[2514]: INFORMACIÓN: Despliegue del directorio
/usr/share/cloudstack-management/webapps/client de la aplicación web
may 16 20:43:21 cloudstack server[2514]: may 16, 2018 8:43:21 PM
org.apache.catalina.loader.WebappClassLoaderBase validateJarFile
may 16 20:43:21 cloudstack server[2514]: INFORMACIÓN:
validateJarFile(/usr/share/cloudstack-management/webapps/client/WEB-INF/lib/servlet-
api-2.5-20081211....ervlet.class
may 16 20:43:21 cloudstack server[2514]: may 16, 2018 8:43:21 PM
org.apache.catalina.loader.WebappClassLoaderBase validateJarFile
may 16 20:43:21 cloudstack server[2514]: INFORMACIÓN:
validateJarFile(/usr/share/cloudstack-management/webapps/client/WEB-INF/lib/servlet-
api-2.5.jar) - ja...ervlet.class
may 16 20:43:22 cloudstack server[2514]: may 16, 2018 8:43:22 PM
org.apache.catalina.loader.WebappClassLoaderBase validateJarFile
may 16 20:43:22 cloudstack server[2514]: INFORMACIÓN:
validateJarFile(/usr/share/cloudstack-management/webapps/client/WEB-INF/lib/tomcat-
embed-core-8.0.15....ervlet.class
Hint: Some lines were ellipsized, use -l to show in full.
```

Por ultimo habilitamos el servicio de Cloudstack para que se inicie automáticamente al iniciar el sistema:

```
[root@cloudstack ~]# systemctl enable cloudstack-management.service
```

Configurando Servidor NFS

Para configurar el almacenamiento de Cloudstack instalaremos el servidor NFS:

```
[root@cloudstack ~]# yum install nfs-utils
```

Creamos los directorios que vamos a utilizar para el almacenamiento de Apache Cloudstack. Un directorio primario para el almacenamiento local de la nube y otro secundario que se emplea para almacenar plantillas, snapshots de las máquinas virtuales e imágenes ISO. El almacenamiento secundario debe estar localizado en la misma zona de disponibilidad que las máquinas huésped a las que sirve. Debe haber exactamente un dispositivo de almacenamiento secundario por cada zona de disponibilidad.

```
[root@cloudstack ~]# mkdir /primary  
[root@cloudstack ~]# mkdir /secondary
```

Ahora necesitamos descomentar los valores de configuración en el archivo /etc/sysconfig/nfs:

```
[root@cloudstack ~]# vim /etc/sysconfig/nfs  
#  
# Define which protocol versions mountd  
# will advertise. The values are "no" or "yes"  
# with yes being the default  
#MOUNTD_NFS_V2="no"  
#MOUNTD_NFS_V3="no"  
#  
#  
# Path to remote quota server. See rquotad(8)  
#RQUOTAD="/usr/sbin/rpc.rquotad"  
# Port rquotad should listen on.
```

```
RQUOTAD_PORT=875
# Optimal options passed to rquotad
#RPCRQUOTADOPTS=""
#
#
# Optional arguments passed to in-kernel lockd
#LOCKDARG=
# TCP port rpc.lockd should listen on.
LOCKD_TCPPORT=32803
# UDP port rpc.lockd should listen on.
LOCKD_UDPPORT=32769
#
#
# Optional arguments passed to rpc.nfsd. See rpc.nfsd(8)
# Turn off v2 and v3 protocol support
#RPCNFSDARGS="-N 2 -N 3"
# Turn off v4 protocol support
#RPCNFSDARGS="-N 4"
# Number of nfs server processes to be started.
# The default is 8.
#RPCNFSDCOUNT=8
# Stop the nfsd module from being pre-loaded
#NFSD_MODULE="noload"
# Set V4 and NLM grace periods in seconds
#
# Warning, NFSD_V4_GRACE should not be less than
# NFSD_V4_LEASE was on the previous boot.
#
# To make NFSD_V4_GRACE shorter, with active v4 clients,
# first make NFSD_V4_LEASE shorter, then restart server.
# This will make the clients aware of the new value.
# Then NFSD_V4_GRACE can be decreased with another restart.
#
# When there are no active clients, changing these values
# can be done in a single server restart.
#
#NFSD_V4_GRACE=90
#NFSD_V4_LEASE=90
#NLM_GRACE_PERIOD=90
#
#
#
# Optional arguments passed to rpc.mountd. See rpc.mountd(8)
#RPCMOUNDOPTS=""
# Port rpc.mountd should listen on.
MOUNTD_PORT=892
#
#
```



```
# Optional arguments passed to rpc.statd. See rpc.statd(8)
#STATDARG=""
# Port rpc.statd should listen on.
STATD_PORT=662
# Outgoing port statd should use. The default is port
# is random
STATD_OUTGOING_PORT=2020
# Specify callout program
#STATD_HA_CALLOUT="/usr/local/bin/foo"
#
#
# Optional arguments passed to rpc.idmapd. See rpc.idmapd(8)
#RPCIDMAPDARGS=""
#
# Set to turn on Secure NFS mounts.
#SECURE_NFS="yes"
# Optional arguments passed to rpc.gssd. See rpc.gssd(8)
#RPCGSSDARGS=""
# Optional arguments passed to rpc.svcgssd. See rpc.svcgssd(8)
#RPCSVCGSSDARGS=""
#
# To enable RDMA support on the server by setting this to
# the port the server should listen on
#RDMA_PORT=20049
```

A continuación, modificaremos el fichero /etc/exports, donde pondremos todos los directorios que queremos que estén disponibles para NFS. Este fichero es importante, para definir permisos y protocolos que podemos usar:

```
[root@cloudstack ~]# vim /etc/exports
/secondary *(rw,async,no_root_squash,no_subtree_check)
/primary *(rw,async,no_root_squash,no_subtree_check)
```

- *rw*: son permisos de lectura y escritura.
- *sync/async*: permite al servidor escribir los datos en el disco cuando lo crea conveniente. Mientras que esto no tiene importancia en un sistema de solo lectura, si una máquina hace cambios en un sistema de ficheros de lectura-escritura y el servidor se cae o se apaga, se pueden perder datos. Especificando la opción *sync*, todas las escrituras en el disco deben hacerse antes de devolver el control al cliente. Esto bajará el rendimiento.
- *no_root_squash*: de forma predeterminada, cualquier solicitud de archivo realizada por el usuario root en la máquina cliente es tratada como el usuario nobody en el servidor. Si se selecciona *no_root_squash*, el root en la máquina

cliente tendrá el mismo nivel de acceso a los archivos del sistema como root en el servidor.

- `no_subtree_check`; permite que no se compruebe la ruta hasta el directorio que se exporta, en el caso de que el usuario no tenga permisos sobre el directorio exportado.

Configuramos el firewall para permitir las conexiones NFS entrantes. Para ello, editamos el fichero `/etc/sysconfig/iptables` y añadimos las siguientes reglas:

```
[root@cloudstack ~]# vim /etc/sysconfig/iptables
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 111 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 111 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 2049 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 32803 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 32769 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 892 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 892 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 875 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 875 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 662 -j ACCEPT
-A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 662 -j ACCEPT
```

Reiniciamos los servicios de iptables:

```
[root@cloudstack ~]# service iptables restart
iptables: Poniendo las cadenas de la política ACCEPT: filte          [ OK ]
iptables: Guardando las reglas del cortafuegos:                  [ OK ]
iptables: Descargando módulos:                                    [ OK ]
iptables: Aplicando reglas del cortafuegos:                      [ OK ]
```

Ahora debemos configurar el servicio nfs para que comience desde el arranque ejecutando los siguientes comandos:

```
[root@cloudstack ~]# service rpcbind start
Iniciando rpcbind:                                               [ OK ]

[root@cloudstack ~]# service nfs start
Inicio de los servicios NFS:                                     [ OK ]
Inicialización de NFS mountd:                                    [ OK ]
Inicialización del demonio NFS:                                  [ OK ]
Iniciando idmapd RPC:                                           [ OK ]

[root@cloudstack ~]# chkconfig rpcbind on
[root@cloudstack ~]# chkconfig nfs on
```


Configuración de plantilla de sistema

CloudStack utiliza una serie de máquinas virtuales del sistema para proporcionar funcionalidad para acceder a la consola de máquinas virtuales, proporcionar diversos servicios de red y administrar diversos aspectos del almacenamiento.

Para añadir las plantillas en nuestro sistemas Cloudstack, debemos descargarla e implementarla en el recurso compartido que acabamos de montar:

```
[root@cloudstack ~]# /usr/share/cloudstack-common/scripts/storage/secondary/cloud-
install-sys-tmplt \
> -m /secondary \
> -u http://cloudstack.apr-get.eu/systemvm/4.6/systemvm64template-4.6.0-kvm.qcow2.bz2 \
> -h kvm -F
--2018-06-04 13:45:14-- http://cloudstack.apr-get.eu/systemvm/4.6/systemvm64template-
4.6.0-kvm.qcow2.bz2
Resolviendo cloudstack.apr-get.eu... 185.27.174.49, 2a00:f10:121:400:403:9cff:fe00:37f
Connecting to cloudstack.apr-get.eu|185.27.174.49|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 319401369 (305M) [application/x-bzip2]
Saving to: '/usr/share/cloudstack-common/scripts/storage/secondary/b898aa62-53ff-4790-
ac34-c04f228b4bd5.qcow2'

100%
[=====
=====
>] 319.401.369 6,66M/s in 26s

2018-06-04 13:45:40 (11,9 MB/s) - '/usr/share/cloudstack-
common/scripts/storage/secondary/b898aa62-53ff-4790-ac34-c04f228b4bd5.qcow2' saved
[319401369/319401369]

Uncompressing to /usr/share/cloudstack-common/scripts/storage/secondary/b898aa62-53ff-
4790-ac34-c04f228b4bd5.qcow2.tmp (type bz2)...could take a long time
Moving to /secondary/template/tmpl/1/3///b898aa62-53ff-4790-ac34-
c04f228b4bd5.qcow2...could take a while
Successfully installed system VM template to /secondary/template/tmpl/1/3/
```

Instalando Hipervisor KVM

Para administrar instancias de KVM en el host, CloudStack usa un agente. Este agente se comunica con el servidor de gestión y controla todas las instancias en el host.

Primero comenzamos por instalar el agente:

```
[root@cloudstack ~]# yum install cloudstack-agent
```

CloudStack usa libvirt para administrar máquinas virtuales. Por lo tanto, es vital que libvirt esté configurado correctamente. Libvirt es una dependencia de cloudstack-agent y ya se encuentra instalado. Para tener una migración en vivo, libvirt tiene que escuchar conexiones TCP no seguras.

Necesitamos editar la configuración de QEMU VNC. Esto se hace editando el fichero /etc/libvirt/qemu.conf y asegurando que la siguiente línea esté presente y sin comentarios.

```
[root@cloudstack ~]# /etc/libvirt/qemu.conf  
vnc_listen = 0.0.0.0
```

Para Configurar libvirt nos dirigimos al fichero siguiente y descomentamos los siguientes parámetros:

```
[root@cloudstack ~]# vim /etc/libvirt/libvirtd.conf  
listen_tls = 0  
listen_tcp = 1  
tcp_port = "16509"  
auth_tcp = "none"  
mdns_adv = 0
```

También descomentamos en el siguiente fichero el parámetro "LIBVIRT_ARGS":

```
[root@cloudstack ~]# vim /etc/sysconfig/libvirtd  
LIBVIRT_ARGS="--listen"
```

A continuación reiniciamos el servicio de libvirt:

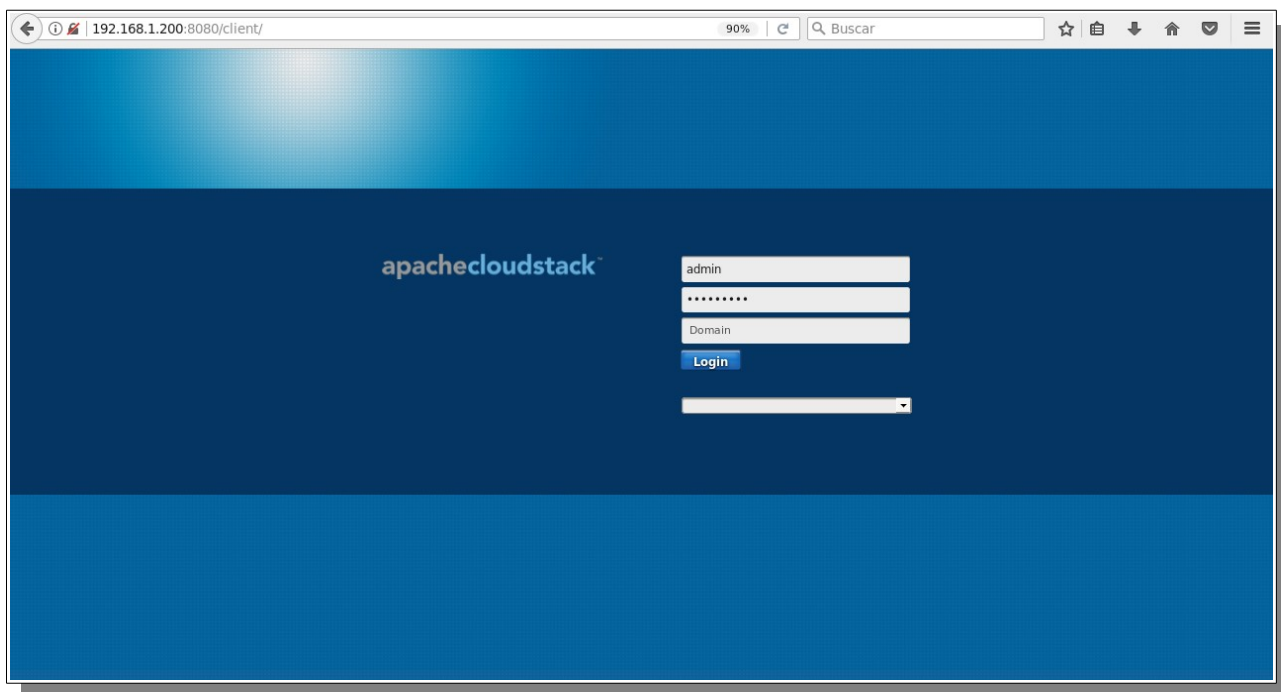
```
[root@cloudstack ~]# service libvirtd restart  
Deteniendo demonio libvirtd: [ OK ]  
Iniciando demonio libvirtd: [ OK ]
```

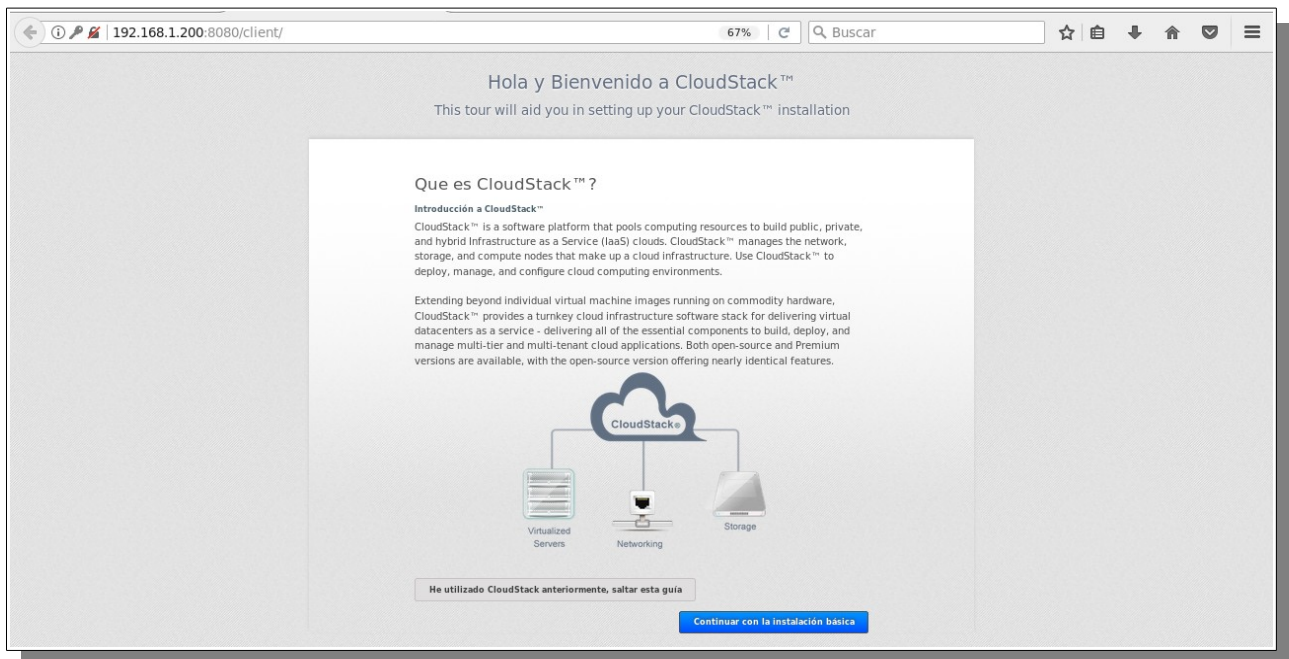
Para completar, verificamos si KVM funciona correctamente en la máquina:

```
[root@cloudstack ~]# lsmod | grep kvm  
kvm_intel      56232  0  
kvm            348526  1 kvm_intel
```

Configurando Apache CloudStack

Para acceder a la interfaz web de CloudStack, simplemente introducimos en el navegador `http://192.168.1.200:8080/client`. El nombre de usuario predeterminado es 'admin' y la contraseña predeterminada es 'password'. Aparecerá una pantalla de bienvenida que nos permite elegir varias opciones para configurar CloudStack. Debemos elegir la opción Continuar con la configuración básica.





En la siguiente pantalla, nos exige que cambiemos la contraseña del usuario administrador.



Usuario: admin

Contraseña: adminroot

Una zona es la entidad de organización más grande en CloudStack, que engloba los diferentes recursos disponibles, cuantas más zonas haya más recursos consume.

- *Diseño de la red pública a utilizar.*
- *El número de pods (red privada) que se desean crear.*
- *El número de clusters que se van a configurar.*
- *El número de hosts o servidores que se van a disponer.*
- *El almacenamiento primario.*
- *El almacenamiento secundario.*

En la siguiente pantalla crearemos una zona con la siguiente información:

- *Nombre: establecemos el nombre 'Zona1' para nuestra nube.*
- *DNS público 1: 8.8.8.8.*
- *DNS público 2: 8.8.4.4.*
- *DNS1 interno: 8.8.8.8.*
- *DNS2 interno: 8.8.4.4.*

Hola y Bienvenido a CloudStack™
This tour will aid you in setting up your CloudStack™ installation

Agregar zona

* Nombre:

* DNS 1:

DNS 2:

* DNS interno una:

DNS interno dos:

Sugerencias
These are DNS servers for use by system VMs in the zone. These DNS servers will be accessed via the private network interface of the System VMs. The private IP address you provide for the pods must

Tras crear la zona, en la siguiente pantalla nos pedirá crear un Pod. En nuestro caso lo crearemos con la siguiente información:

- *Nombre: usaremos Pod1 para nuestra nube.*
- *Gateway: usaremos 192.168.1.1 como nuestra puerta de enlace*
- *Netmask: 255.255.255.0*

- *Rango de las IP del sistema: 192.168.1.201 – 192.168.1.254*

Hola y Bienvenido a CloudStack™
This tour will aid you in setting up your CloudStack™ installation

Añadir Pod

* Nombre:

* puerta de enlace:

* máscara de red:

* Rango de IP:

Sugerencias

Este es el rango de direcciones IP en la red privada que la CloudStack utiliza para administrar las VMs del Almacenamiento Secundario y consola Proxy. Estas direcciones IP se han tomado de la misma subred que los servidores informáticos.

A continuación, creamos la red de invitados con la siguiente información:

- *Puerta de enlace de invitado: 192.168.1.1*
- *Máscara de red de invitado: 255.255.255.0*
- *IP de inicio / finalización de invitado: 192.168.1.180 – 192.168.1.198*

Hola y Bienvenido a CloudStack™
This tour will aid you in setting up your CloudStack™ installation

Agregar red de invitado

* puerta de enlace:

* máscara de red:

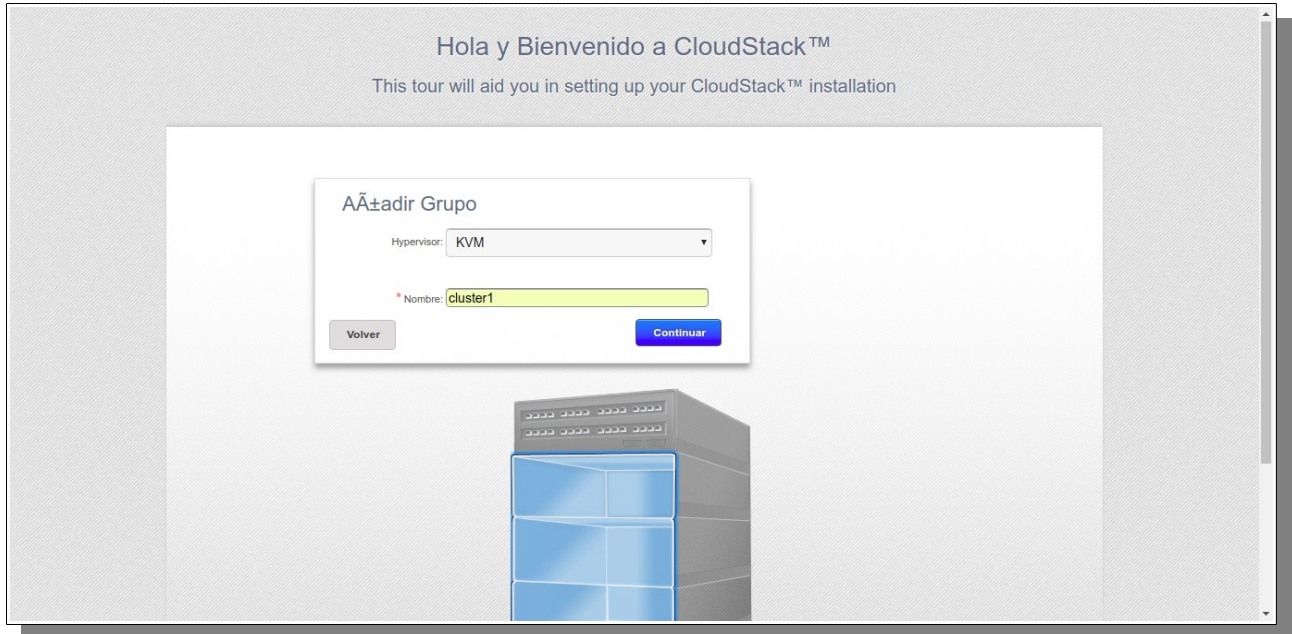
* Rango de IP:

Sugerencias

The range of IP addresses that will be available for allocation to guests in this zone. If one NIC is used, these IPs should be in the same CIDR as the pod CIDR.

Una vez agregada una zona, solo necesitamos añadir algunos elementos más para configurar el cluster.

- *Nombre: Cluster1*
- *Hipervisor: KVM*



Ahora agregamos el primer host a nuestro cluster:

- *Nombre de host: usaremos una dirección IP, en este caso, 192.168.1.200, ya que no tenemos configurado un servidor DNS para resolver por nombres.*
- *Nombre de usuario: root.*
- *Contraseña: contraseña del sistema operativo para el usuario raíz (adminroot).*



Una vez, configurado el cluster, elegimos NFS como tipo de almacenamiento y luego ingresamos los siguientes valores en los campos:

- Nombre: primary1
- Servidor: 192.168.1.200
- Ruta: definimos la ruta donde se encuentra el directorio a montar, /primary

Hola y Bienvenido a CloudStack™
This tour will aid you in setting up your CloudStack™ installation

Añadir Almacenamiento primario

* Nombre:

Protocolo:

Alcance:

* Servidor:

* Ruta:

Sugerencias
(para NFS) En NFS este es el directorio exportado desde el servidor. Directorio (por SharedMountPoint). Con KVM este es el directorio de cada host en donde se monta el almacenamiento.

En la siguiente pantalla nos pedirá configurar el almacenamiento secundario. Introducimos los siguientes valores.

- Servidor: Usaremos la dirección IP 192.168.1.200
- Ruta: /secondary

Hola y Bienvenido a CloudStack™
This tour will aid you in setting up your CloudStack™ installation

Añadir secundaria almacenamiento

* servidor NFS:

Provider:

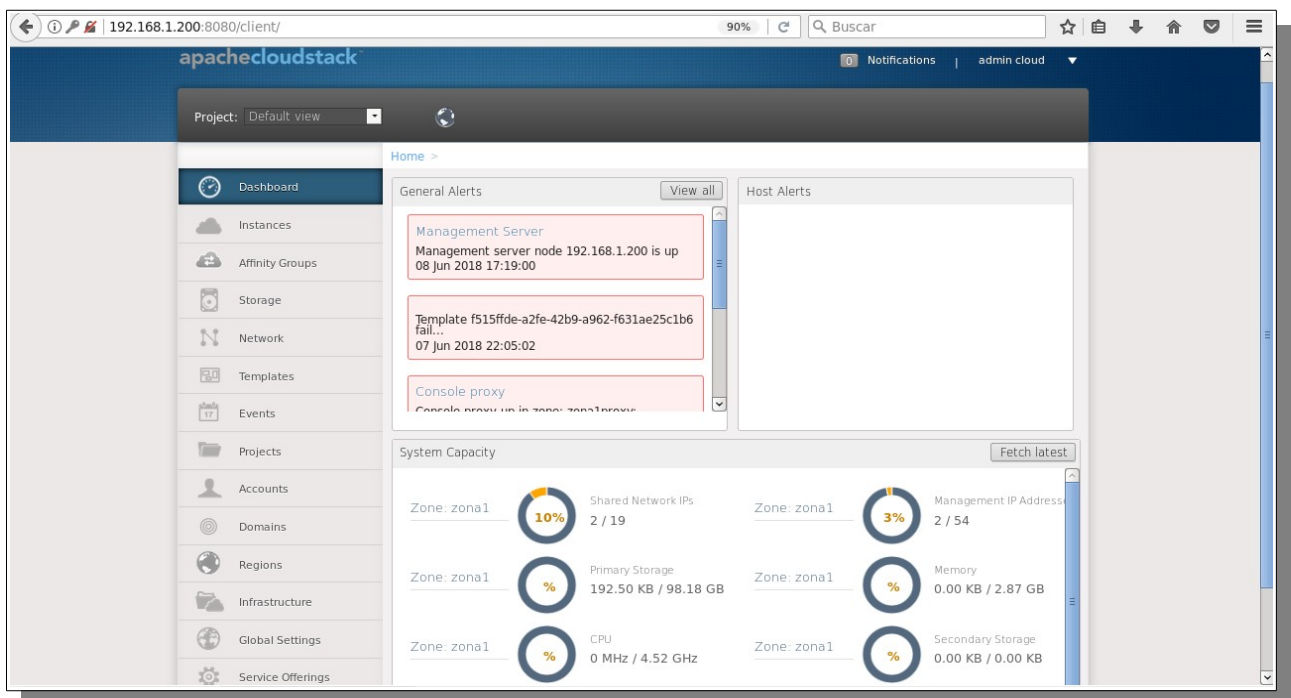
* Ruta:

Sugerencias
El path exportado, ubicado en el servidor especificado anteriormente

Tras añadir y configurar los elementos necesarios para crear la zona, ya podemos pulsar en lanzar y acceder al panel de administración de cloudstack.



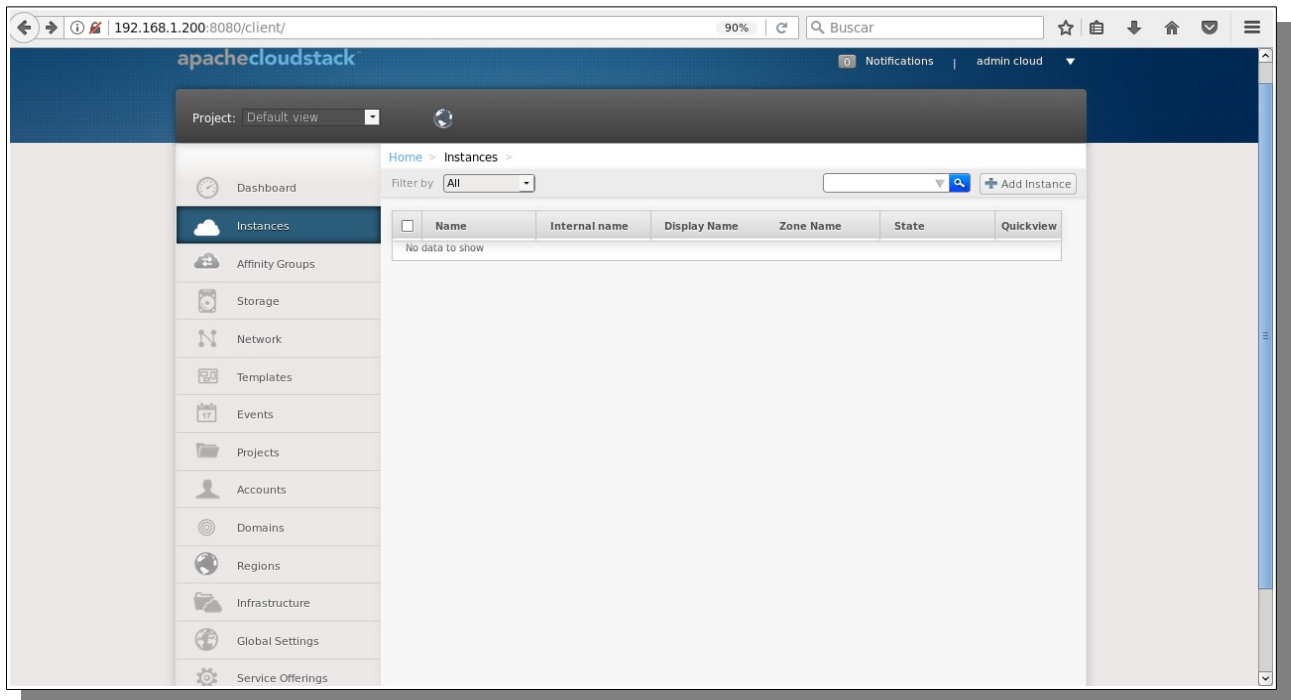
Si todo ha ido correctamente, nos aparecerá la siguiente ventana, donde podemos administrar nuestra nube Cloudstack,



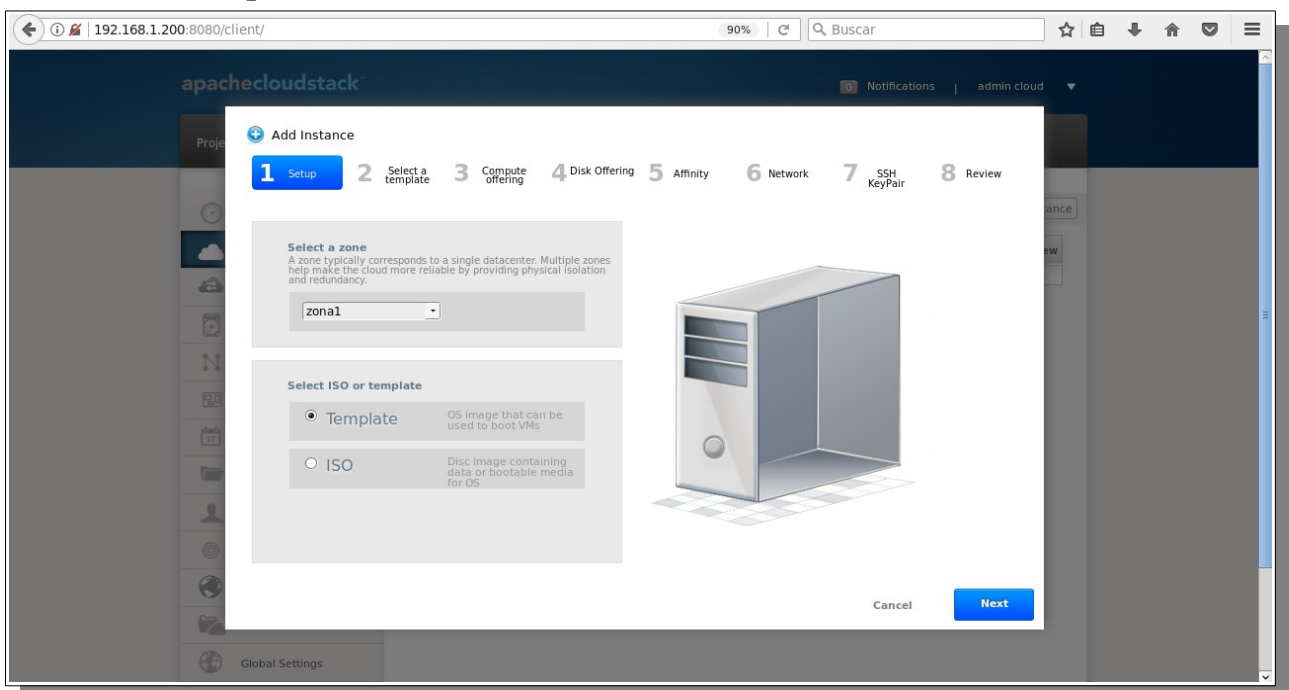
Explorando Apache CloudStack

Creación de Máquinas virtuales:

Para crear una instancia nos situamos en el panel de administración en la pestaña instances y pulsamos en el botón +Add instance:



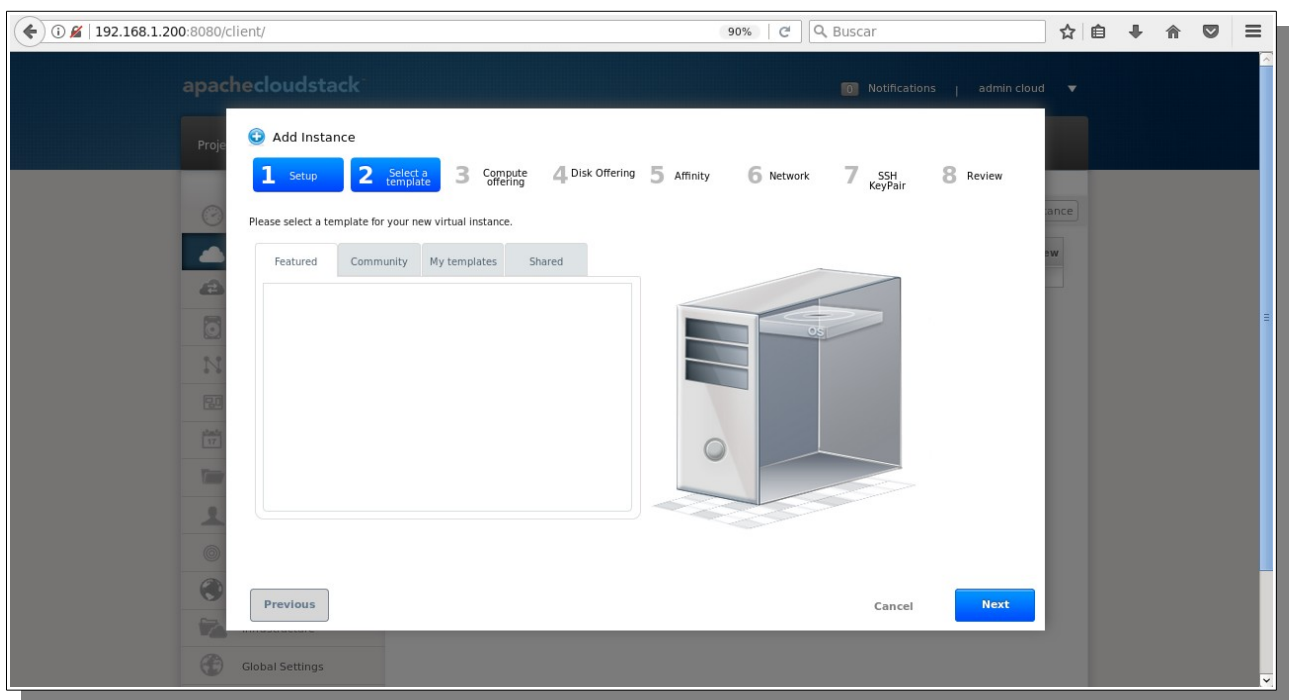
Seleccionamos la zona, que en este caso, solo hay una (la que hemos creado anteriormente) y seleccionamos, también, si la instancia se creará desde una imagen ISO o desde una plantilla.



Se puede crear una instancia vacía y proceder a instalar el sistema operativo desde una ISO que esté disponible en Cloudstack (listado de ISOS) o crear directamente la instancia a partir de un template. Un template es una plantilla predefinida ya configurada a partir de la que se crean instancias (listado de templates).

Hay 4 categorías de Templates o ISOs:

- *Featured*: creadas por el administrador para todos los usuarios.
- *Community*: creadas por otros usuarios y hechas públicas.
- *My templates*: creadas por el usuario y privadas.
- *Shared*: compartidas por otros usuarios.



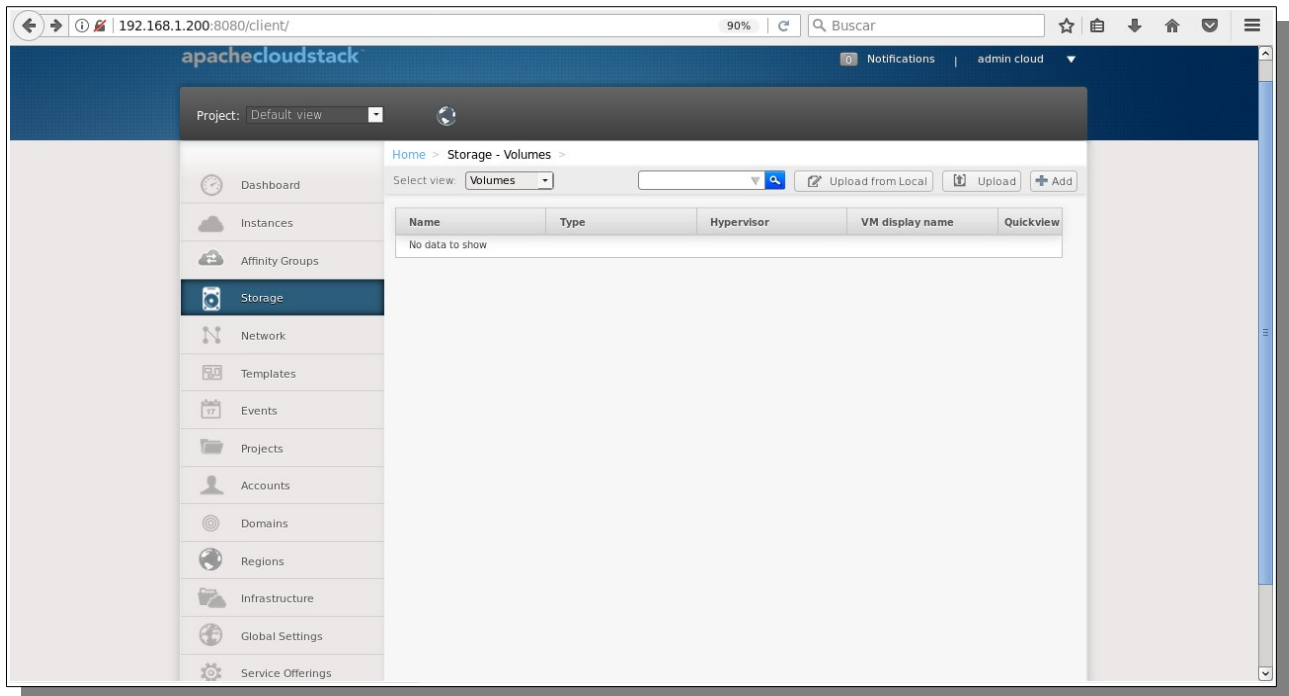
En la pestaña “Compute offering” hay que elegir la configuración física, número de procesadores virtuales, megahercios de cada procesador y cantidad de MegaBytes de memoria RAM. “HA” quiere decir alta disponibilidad, por lo que, si la instancia se apaga por la razón que sea Cloudstack la inicia de nuevo de forma automática. Por último, seleccionamos el tamaño del disco en la pestaña “Disk offering” y la red en la pestaña “Network”. Con estos datos ya podemos generar una instancia.

Nota:

Por un error en el almacenamiento secundario de apache cloudstack nos impide crear una instancia, ya que las imágenes y templates se almacenan ahí, con lo que la lista de imágenes disponibles es nula.

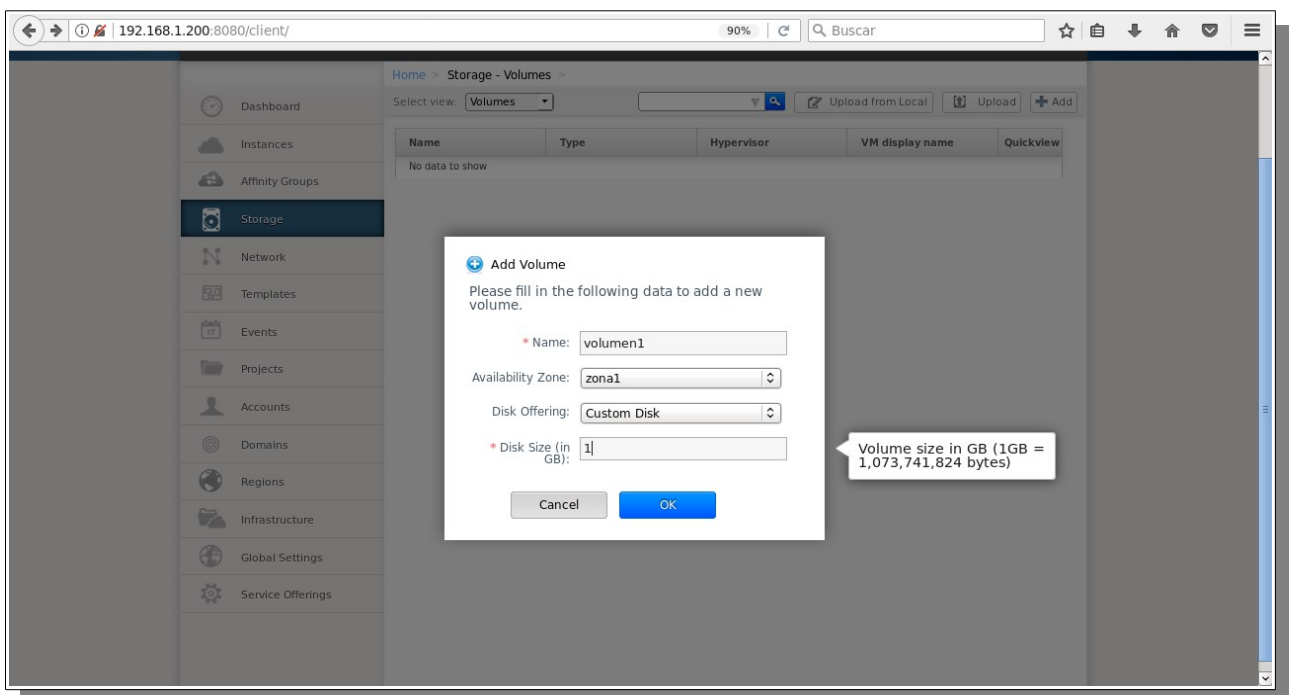
Creación de Espacios de Almacenamiento:

Para crear un espacio de almacenamiento externo para una máquina virtual, nos situamos en el panel de administración de cloudstack en la pestaña "Storage".

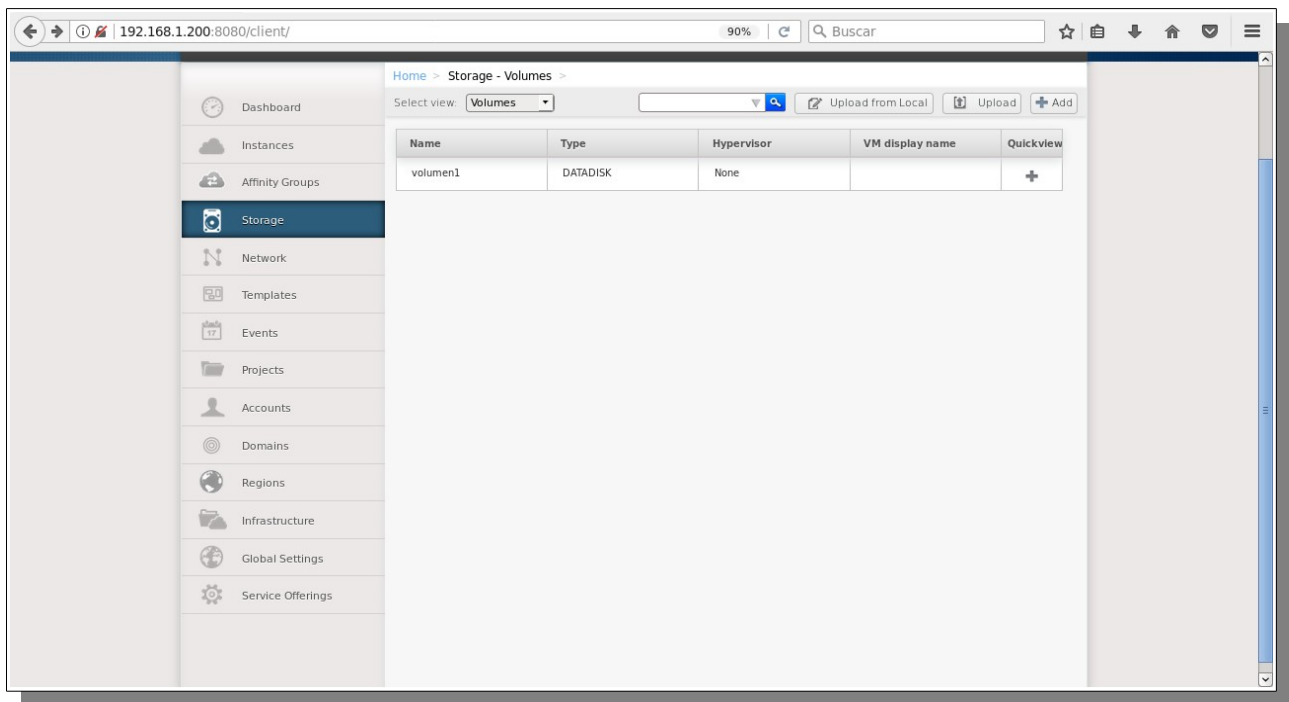


Pulsamos en el botón + Add y rellenamos los siguiente información:

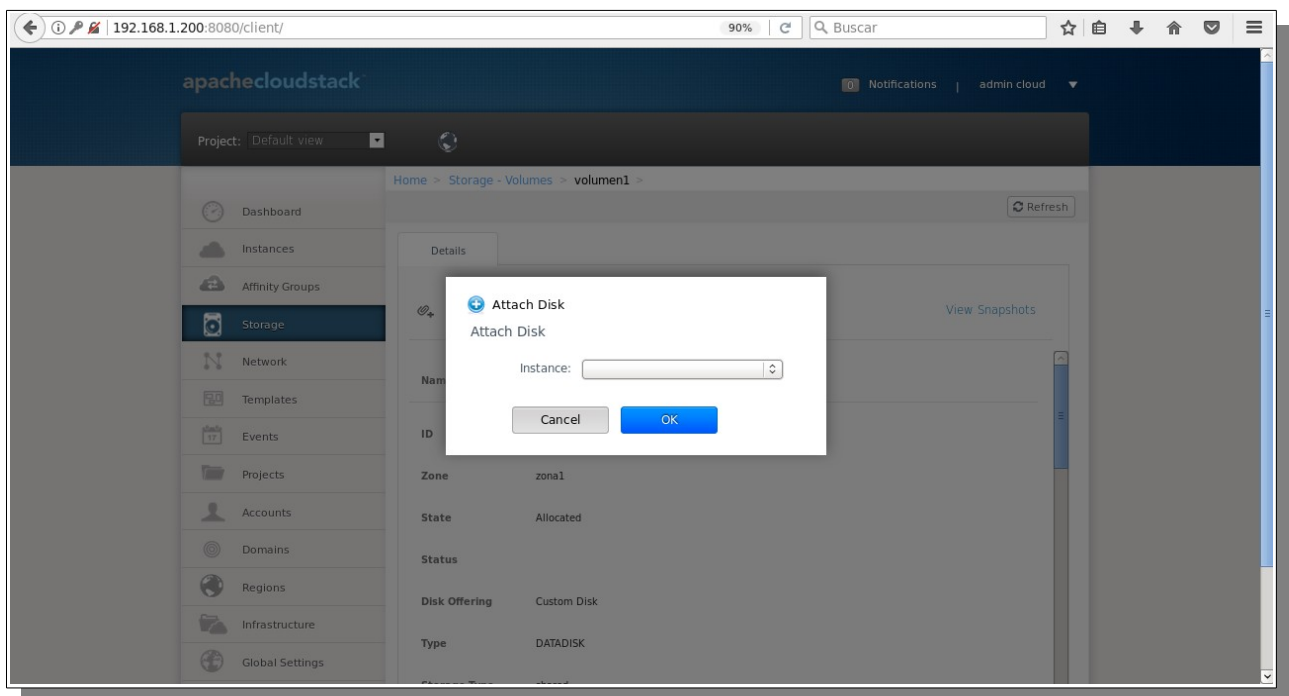
- *Nombre: volumen1*
- *Zona: zona1*
- *Tamaño: 1 GB*



Comprobamos como se ha generado el volumen correctamente.

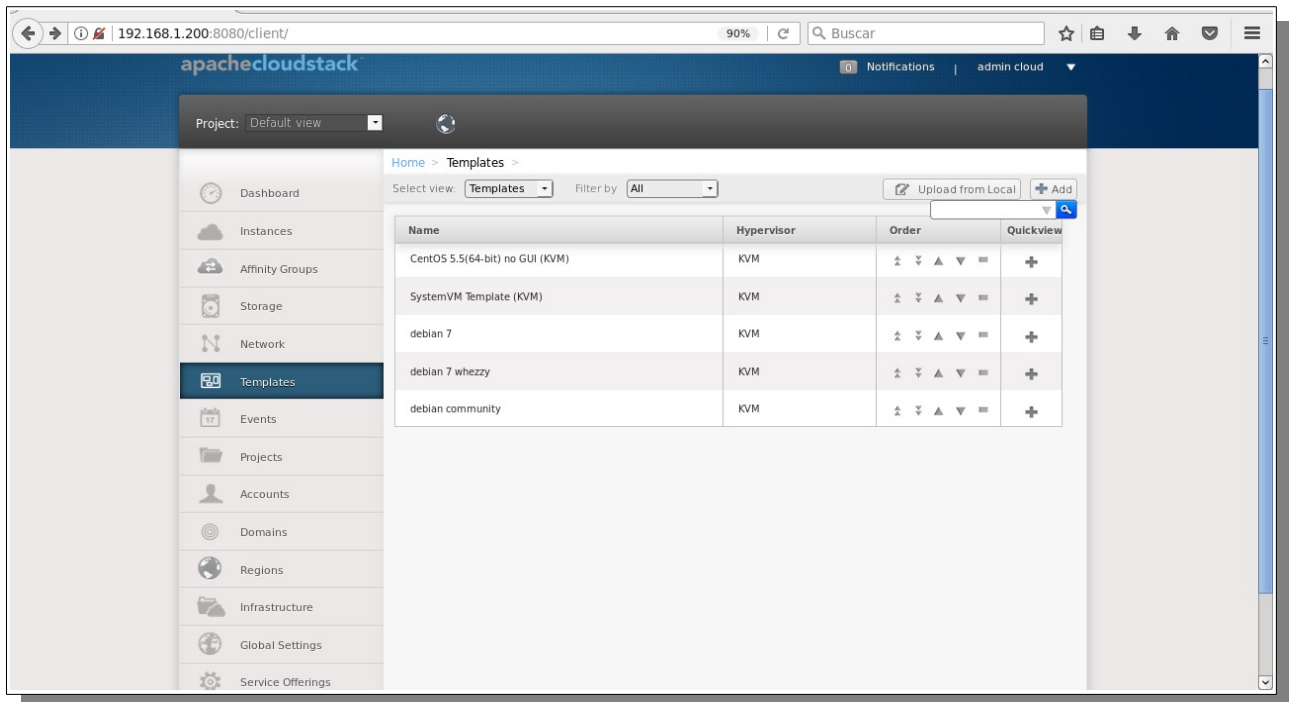


Para vincular un volumen a una instancia pulsamos sobre el volumen, pulsamos en el icono attach y seleccionamos la instancia a la que le queremos vincular la imagen.



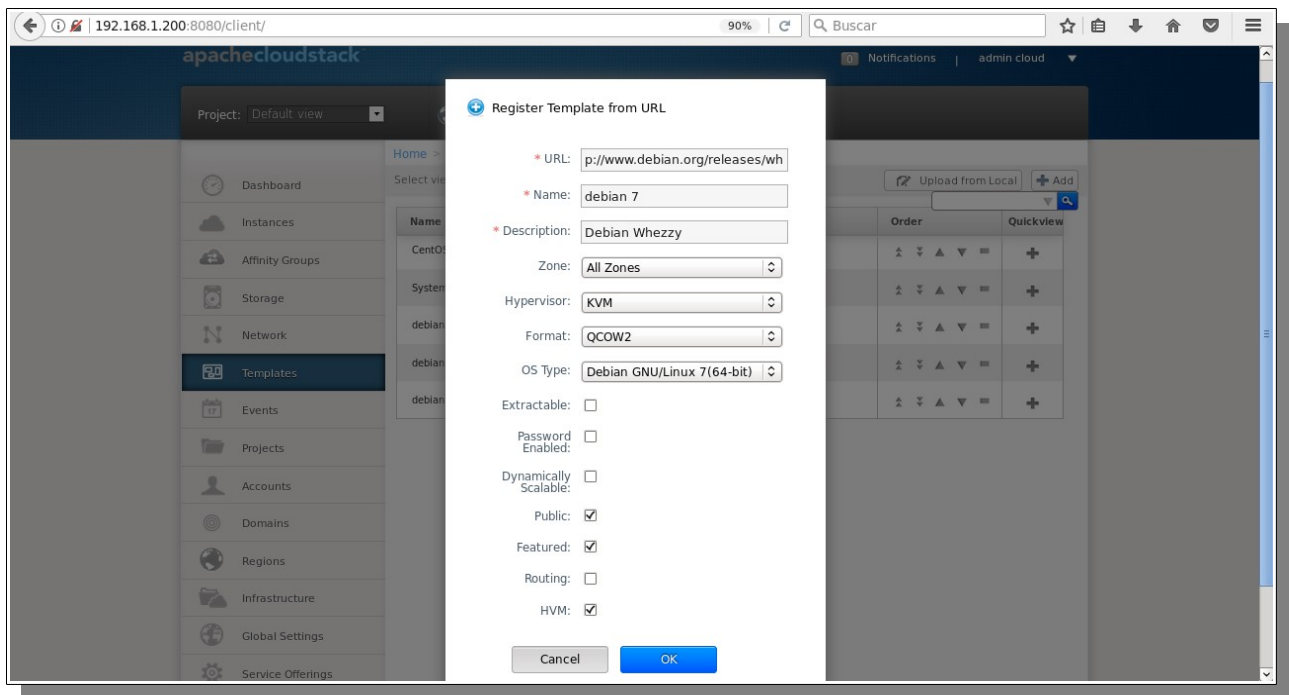
Subida de Imágenes:

Para añadir una nueva imagen o template nos ubicamos en la pestaña "Templates" del panel de administración.



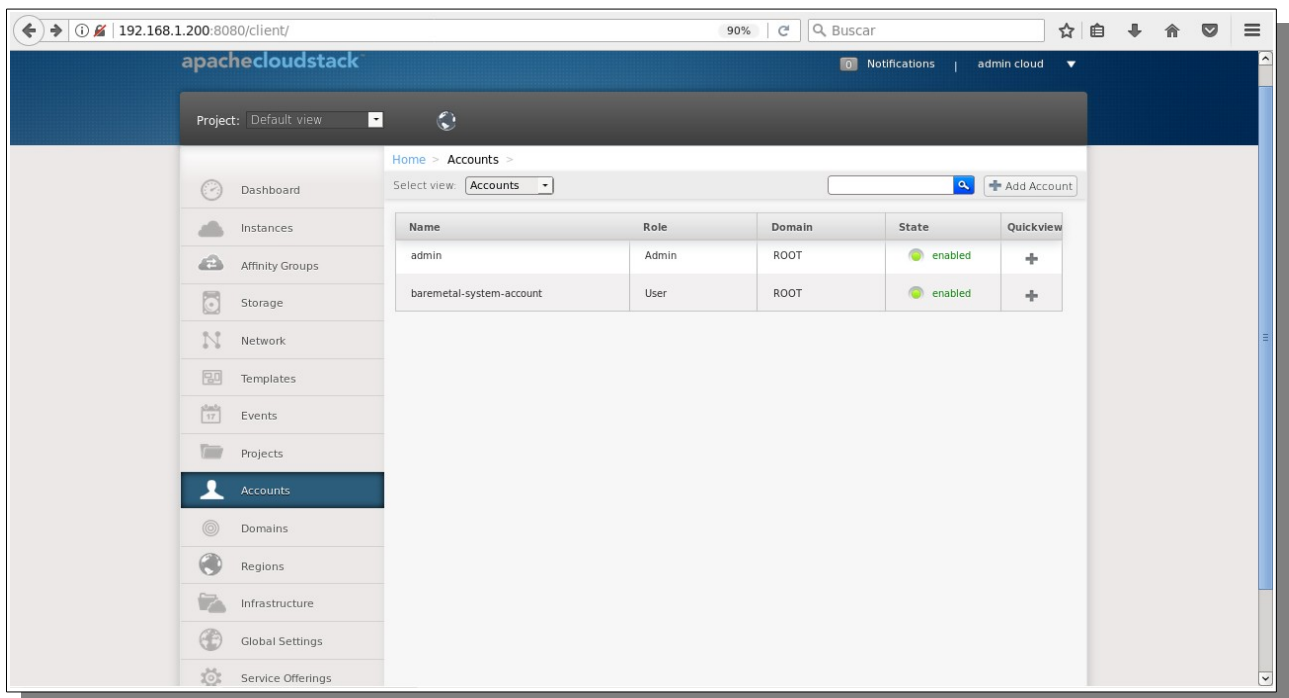
Pulsamos en el botón + Add para añadir una nueva imagen y rellenamos la siguiente información:

- *URL: <http://www.debian.org/releases/wheezy>*
- *Nombre: debian 7*
- *Descripción: debian wheezy*
- *Zona: zona1*
- *Hipervisor: KVM*
- *Formato: qcow2*
- *Sistema operativo: debian 7 64bit*



Gestión de Usuarios:

Para añadir un nuevo usuario nos ubicamos en la pestaña “Accounts” del panel de administración.



Pulsamos en el botón + Add Account para agregar un nuevo usuario y rellenos la siguiente información.

The screenshot shows the 'Add Account' modal form in the OpenStack dashboard. The form is titled 'Add Account' and contains the following fields:

- * Username: elizabeth
- * Password: [masked]
- * Confirm password: [masked]
- * Email: h.morbeg@gmail.com
- * First Name: Elizabeth
- * Last Name: Moreno
- * Domain: ROOT (dropdown)
- Account: [empty]
- * Type: User (dropdown)
- Timezone: Europe/Brussels (dropdown)
- Network Domain: [empty]

At the bottom of the modal are 'Cancel' and 'Add' buttons. The background shows the 'Accounts' page with a table of existing accounts.

Name	Role	Domain	State	Quickview
admin	Admin	ROOT	enabled	+
baremetal-system-account	User	ROOT	enabled	+
elizabeth	User	ROOT	enabled	+

Comprobamos que se ha generado el usuario correctamente.

The screenshot shows the 'Accounts' page in the OpenStack dashboard. The 'Accounts' tab is selected, and the table below shows the list of accounts. The newly added user 'elizabeth' is now listed.

Name	Role	Domain	State	Quickview
admin	Admin	ROOT	enabled	+
baremetal-system-account	User	ROOT	enabled	+
elizabeth	User	ROOT	enabled	+

Creación de recursos para instancias:

Para crear un nuevo recurso personalizado, también llamado sabores, para nuestras instancias nos ubicamos en la pestaña "Service Offering" y pulsamos en el botón +Add compute offering.

Rellenamos la siguiente información:

- Nombre: big
- Descripción: big instance
- CPU cores: 2
- Memoria RAM: 1024

The screenshot shows the 'Add compute offering' form in the Apache CloudStack interface. The form is filled with the following values:

- Name: big
- Description: big instance
- Storage Type: shared
- Provisioning Type: thin
- Custom: ☐
- # of CPU Cores: 2
- CPU (in MHz): 2
- Memory (in MB): 1024
- Network Rate (MB/s):
- QoS Type:
- Offer HA: ☐
- Storage Tags:
- Host Tag:
- CPU Cap: ☐
- Public: ☐
- Volatile: ☐
- Deployment planner:
- GPU:
- Domain: ROOT

The form has 'Cancel' and 'OK' buttons at the bottom. In the background, the 'Offerings' table is visible with columns: Description, Order, and Quickview. It lists 'Medium Instance' and 'Small Instance'.

En la siguiente imagen comprobamos que se ha generado correctamente.

The screenshot shows the Apache CloudStack web interface. The browser address bar indicates the URL is 192.168.1.200:8080/client/. The interface has a dark blue header with the Apache CloudStack logo and a navigation bar. A sidebar on the left contains various navigation options, with 'Service Offerings' currently selected. The main content area displays the 'Compute Offerings' page, which includes a table of offerings.

Name	Description	Order	Quickview
big	big instance	1	+
Medium Instance	Medium Instance	2	+
Small Instance	Small Instance	3	+

Bibliografía

http://copro.com.ar/Apache_CloudStack.html

<https://www.shapeblue.com/apache-cloudstack/>

<https://cloudstack.apache.org/>

<http://www.datamation.com.ar/nubes-privadas-openstack-o-cloudstack-4367>

<http://docs.cloudstack.apache.org/projects/cloudstack-installation/en/4.6/>

<https://whatis.techtarget.com/definition/CloudStack>

<https://semillerothmccsl.files.wordpress.com/2016/11/anexo2.pdf>

<https://discussions.citrix.com/topic/352227-console-proxy-or-system-vm-failed-to-start/>

<http://users.cloudstack.apache.narkive.com/IjAGr29l/secondary-storage-problem>

[https://stackoverflow.com/questions/25324799/secondary-storage-not-recognized-in-apache-cloud-stack?](https://stackoverflow.com/questions/25324799/secondary-storage-not-recognized-in-apache-cloud-stack?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa)

<https://reportedigital.com/cloud/que-es-nube-cloudstack-cuales-son-sus-ventajas/>

<https://www.youtube.com/watch?v=WbBTDUMk0w0>