

---

OpenStack soporta un buen número de tecnologías para proporcionar almacenamiento de bloques, en este caso vamos a explicar la utilización de LVM + iSCSI que es la tecnología utilizada por defecto y quizás la más adecuada para explicar el funcionamiento de OpenStack gracias a su sencillez. A diferencia del caso de KVM, que es sin duda ninguna el hipervisor más utilizado en despliegues de OpenStack, LVM + iSCSI no es la combinación más extendida ya que hay múltiples opciones (Ceph RBD, GlusterFS, sistemas de almacenamiento de múltiples fabricantes, etc.), de entre todas es Ceph RBD la opción preferida de acuerdo a la última [encuesta de uso de OpenStack](#).

## LVM

Logical Volume Manager 2 (LVM2) o simplemente LVM es una implementación de volúmenes lógicos para el kernel de linux. Permite abstraer los dispositivos de bloques lógicos utilizados por el sistema o los usuarios de los dispositivos físicos de la máquina, permitiendo una gestión más adecuada y dinámica de los recursos.

LVM en sí mismo no proporciona redundancia, ésta habitualmente se proporciona utilizando RAID por debajo.

## iSCSI

iSCSI es un estándar que permite utilizar la interfaz de transferencia de datos de bus SCSI a través de una red TCP/IP. Es un protocolo de la capa de transporte que básicamente permite la utilización de un dispositivo de bloques en un equipo remoto.

# Creación de un volumen

Utilizamos el cliente cinder para crear un nuevo volumen:

```
$ cinder create --display-name vol3 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-05-29T09:54:39.441869
display_description	None
display_name	vol3
id	d61284c3-4daa-48e1-9731-cd6d4361b4c6
metadata	{}
size	1
snapshot_id	None
source_volid	None
status	creating
volume_type	None

Desde el nodo de almacenamiento (equipo en el que se encuentra instalado cinder o que está configurado como backend) veremos la creación de un nuevo volumen lógico del tamaño solicitado:

```
# lvs
...
volume-d61284c3-4daa-48e1-9731-cd6d4361b4c6    cinder-volumes -wi-ao----    1,00g
...
```

Este volumen se ha creado sobre el grupo de volúmenes *cinder-volumes* y se identifica con el nombre *volume-* seguido del UUID del volumen de OpenStack.

Aunque el volumen ya esté creado, el target iSCSI correspondiente no se creará hasta que se realice la asociación del volumen a una determinada instancia.

## Asociación de un volumen a una instancia

Para asociar un volumen a una instancia utilizamos el cliente nova:

```

$ nova volume-attach f69ab4d9-ed28-406e-8a78-67f62a5e4bc2 d61284c3-4daa-48e1-9731-cd6d4361
b4c6
+-----+-----+
| Property | Value |
+-----+-----+
| device   | /dev/vdb |
| id       | d61284c3-4daa-48e1-9731-cd6d4361b4c6 |
| serverId | f69ab4d9-ed28-406e-8a78-67f62a5e4bc2 |
| volumeId | d61284c3-4daa-48e1-9731-cd6d4361b4c6 |
+-----+-----+

```

En ese momento se crea un nuevo target iSCSI asociado a este volumen lógico que podemos ver con la siguiente instrucción:

```

# tgtadm --lld iscsi --op show --mode target
...
Target 7: iqn.2010-10.org.openstack:volume-d61284c3-4daa-48e1-9731-cd6d4361b4c6
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
  LUN information:
    LUN: 0
      Type: controller
        SCSI ID: IET      00070000
        SCSI SN: beaf70
        Size: 0 MB, Block size: 1
        Online: Yes
        Removable media: No
        Prevent removal: No
        Readonly: No
        Backing store type: null
        Backing store path: None
        Backing store flags:
    LUN: 1
      Type: disk
        SCSI ID: IET      00070001

        SCSI SN: beaf71
        Size: 1074 MB, Block size: 512
        Online: Yes
        Removable media: No
        Prevent removal: No
        Readonly: No
        Backing store type: rdwr
        Backing store path: /dev/cinder-volumes/volume-d61284c3-4daa-48e1-9731-
cd6d4361b4c6
        Backing store flags:
  Account information:
  ACL information:
    ALL

```

Y en el nodo de computación correspondiente podemos ver la conexión realizada con detalle:

```
# iscsiadm -m session -P 3
Target: iqn.2010-10.org.openstack:volume-d61284c3-4daa-48e1-9731-cd6d4361b4c6 (non-flash)
  Current Portal: 192.168.222.11:3260,1
  Persistent Portal: 192.168.222.11:3260,1
  *****
  Interface:
  *****
  Iface Name: default
    Iface Transport: tcp
    Iface Initiatorname: iqn.1993-08.org.debian:01:cab427965c1
    Iface IPaddress: 192.168.222.23
    Iface HWaddress: <empty>
    Iface Netdev: <empty>
    SID: 121
    iSCSI Connection State: LOGGED IN
    iSCSI Session State: LOGGED_IN
    Internal iscsid Session State: NO CHANGE
  *****
  Timeouts:
  *****
  Recovery Timeout: 120
  Target Reset Timeout: 30
  LUN Reset Timeout: 30
  Abort Timeout: 15
  *****
  CHAP:

  *****
  username: ZQg9H5oC5b5rHBthfSFf
  password: *****
  username_in: <empty>
  password_in: *****
  *****
  Negotiated iSCSI params:
  *****
  HeaderDigest: None
  DataDigest: None
  MaxRecvDataSegmentLength: 262144
  MaxXmitDataSegmentLength: 8192
  FirstBurstLength: 65536
  MaxBurstLength: 262144
  ImmediateData: Yes
  InitialR2T: Yes
  MaxOutstandingR2T: 1
  *****
  Attached SCSI devices:
  *****
  Host Number: 127          State: running
  scsi127 Channel 00 Id 0 Lun: 0

  scsi127 Channel 00 Id 0 Lun: 1
          Attached scsi disk sdj          State: running
```

Si volcamos la salida del “kernel ring buffer” con la instrucción dmesg del nodo de computación donde se encuentra ejecutándose la instancia, vemos una salida como la siguiente:

```
scsi5 : iSCSI Initiator over TCP/IP
scsi 5:0:0:0: RAID IET Controller 0001 PQ: 0 ANSI: 5
scsi 5:0:0:0: Attached scsi generic sg7 type 12
scsi 5:0:0:1: Direct-Access IET VIRTUAL-DISK 0001 PQ: 0 ANSI: 5
sd 5:0:0:1: Attached scsi generic sg8 type 0
sd 5:0:0:1: [sdj] 2097152 512-byte logical blocks: (1.07 GB/1.00 GiB)
sd 5:0:0:1: [sdj] Write Protect is off
sd 5:0:0:1: [sdj] Mode Sense: 49 00 00 08
sd 5:0:0:1: [sdj] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
sdg: unknown partition table
sd 5:0:0:1: [sdj] Attached SCSI disk
```

Es decir, el que realiza la conexión iSCSI es el nodo de computación y el dispositivo de bloques remoto aparece como un dispositivo de bloques local asociado a /dev/sdj. Sin embargo este dispositivo de bloques no lo utiliza directamente el nodo de computación sino que se lo transfiere a la máquina virtual modificando el fichero /etc/libvirt/qemu/instance-0000001c.xml con las líneas:

```
<disk type='block' device='disk'>
  <driver name='qemu' type='raw' cache='none' />
  <source dev='/dev/disk/by-path/ip-192.168.0.68:3260-iscsi-iqn.2010-10.org.openstack:volu
me-d61284c3-4daa-48e1-9731-cd6d4361b4c6-lun-1' />
  <target dev='vdb' bus='virtio' />
  <serial>d61284c3-4daa-48e1-9731-cd6d4361b4c6</serial>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
</disk>
```

## Instantáneas

Si hacemos una instantánea de un volumen:

```
$ cinder snapshot-create vol3
+-----+-----+
| Property | Value |
+-----+-----+
| created_at | 2014-05-29T10:39:24.806930 |
| display_description | None |
| display_name | None |
| id | 9236ca13-0b71-4050-8ac0-467827719327 |
```

metadata	{}
size	1
status	creating
volume_id	d61284c3-4daa-48e1-9731-cd6d4361b4c6

Se utiliza directamente una instantánea de LVM:

```
# lvs
LV          VG          Attr      LSize   Pool Origin
_snapshot-3546a39f-fc32-42ea-9c51-2d6959f9798a cinder-volumes swi-a-s--- 1,00g   vol
ume-70ba58bf-1201-4b27-8de3-a1f2f1bf49ed    0,00
```

Pero no se ha creado ningún target nuevo ya que las instantáneas de volumen no son directamente utilizables como volúmenes sino que es necesario crear un nuevo volumen desde ellas para poder asociarlo posteriormente a una instancia.

## Borrado seguro de volúmenes

Puesto que estamos en un sistema de recursos compartidos el mecanismo que utiliza por defecto OpenStack para borrar los volúmenes es seguro porque antes de eliminar el volumen lógico procede a llenarlo de ceros para evitar que los datos que allí hubieran los pueda leer posteriormente otro usuario.

Para ver este comportamiento bien es preciso tener configurado cinder en modo debug y proceder a eliminar un volumen cualquiera:

```
$ cinder delete 70ba58bf-1201-4b27-8de3-a1f2f1bf49ed
```

Si vemos los registros de cinder-volume podemos extraer los siguientes mensajes:

```
2014-05-29 12:50:26.023 13973 DEBUG cinder.openstack.common.processutils [req-1f17dc86-1de6-4483-b106-d09cd2ab70be 0f94f0bbcdee4715ba2750502bb0d63f 0912c80bef254d7b9352632793cf75b9] Running cmd (subprocess): sudo cinder-rootwrap /etc/cinder/rootwrap.conf dd count=0 if=/dev/zero of=/dev/mapper/cinder--volumes-volume--70ba58bf--1201--4b27--8de3--a1f2f1bf49ed iflag=direct oflag=direct execute /usr/lib/python2.6/site-packages/cinder/openstack/common/processutils.py:142
...
2014-05-29 12:50:43.121 13973 INFO cinder.volume.manager [req-1f17dc86-1de6-4483-b106-d09cd2ab70be 0f94f0bbcdee4715ba2750502bb0d63f 0912c80bef254d7b9352632793cf75b9] ... 70ba58bf-1201-4b27-8de3-a1f2f1bf49ed
```

qZad/00e 0T94T000C0ee4/150a2/5050Z000063T 0912C800eT2540/0935Z63Z/93CT/509] volume /00a580  
f-1201-4b27-8de3-a1f2f1bf49ed: deleted successfully

---