

x5

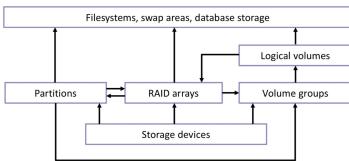
T8 : Almacenamiento

Almacenamiento Local

Sistema de Ficheros (Filesystem)

- ↳ Sist. op.  organiza jerarquía de ficheros reportados sobre los ds. dispositivos de almacenamiento disponible.
- Mantiene la jerarquía de ficheros reportados sobre los ds. dispositivos de almacenamiento disponible.
- contempla los datos (archivos)
- metadatos (catalogos)
- ? Sist. Unix → maneja solo jerarquía de directorios
- cuantos sub-ds de directorios deberán → ? punto de mount para que sea accesible local → remoto
- Nombre →

Gestión del Almacenamiento → Niveles



Dispositivo de Almacenamiento Proporciona acceso al usuario a niv. de bloques → Sist. de Almacenamiento → niv. de bloques → Sist. de Archivos → Sist. de Archivos → Sist. de Archivos → ...

Partición: porción de longitud fija del disp. de Almacenamiento

- ↳ cada partición < formato Sist. de Archivos (formato)
- comporta como unidad de Almacenamiento



RAID (Redundant Array of Inexpensive/Independent Disk)

combina múltiples disp. de Almacenamiento → formar nuevo disp. virtual → aumentar rendimiento → fiabilidad, gastos

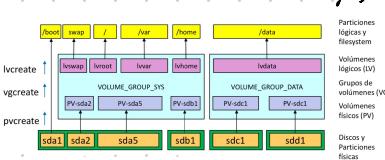


Volumenes lógicos y Grupos de Volumenes:

LVM, logical Volume Manager

- ↳ Sist. de gestión de Almacenamiento q. permite agregar dispositivos (pool of storage) para formar

Grupos de volumenes lógicos



En UNIX todo es un fichero, y sólo, un proceso!

Filesystem → Archivos de org. en Particiones lógicas

Hay muchos Sist. de Archivos de Linux y Aliados

Linux: ext2fs, ext3fs, ext4fs, Reiserfs, ...

Aliados: jfs (IBM), xfs (SGI), nfs y pfs (Mac)...

JOURNALING

Registro o diario de transacciones

cambios realizados en el Sist. de Archivos antes de actualizar los datos. Suministra Almacenamiento

Facilita recuperación rápida del Sist. después una caída
Cada vez q. se monta el Sist. Archivos → revisa registro → realizar tareas pendientes

↳ lo incorporan todos los Sist. Archivos modernos: ext3, ext4, xfs, ...

Comandos Básicos

lsblk, df (unidades de disco presentes y espacio de disco usado)
Opciones: -h (unidades a escala), -i (inode o nodos índice), -T (tipo)

fdisk (gestión de particiones)

Opciones: -l (listar), interactivo (sin opciones); m (ayuda), p (listado particiones), d (borrar partición), n (nueva partición), t (tipo), a (partición de arranque), w (salvar cambios) ... Alternativa: parted

mkfs. <tipo_sistema>, mkfs -t <tipo_sistema>, mkreiser (formatear partición). Opciones: -m porcentaje, -c (check)...

fscck (chequear una partición)

Opciones: -A (todos los filesystem de fstab), -C (progreso), ...

du (espacio consumido por directorios)

Opciones: -a (también archivos), -h (unidades a escala), -s (resumen)

mount/umount (montar el sistema de archivos). Montajes permanentes: /etc/fstab

sistema de Archivos

RAID Software

mdadm (Multiple Device Administrator) permite administrar diferentes configuraciones RAID software (paquete mdadm).

Modos de trabajo: linear (combina discos de tamaños arbitrarios), raid 0 (stripe), raid 1 (mirror), raid 4, raid 5, ... Ejemplos:

mdadm --create --verbose /dev/md0 --level=0
--raid-devices=2 /dev/sdb /dev/sdc

mdadm --create --verbose /dev/md1 --level=1
--raid-devices=2 /dev/sdb /dev/sdc

mdadm --create --verbose /dev/md2 --level=5
--raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
--spare-devices=1 /dev/sde

Para ver el estado del dispositivo RAID:

mdadm --detail /dev/md0

Para eliminar un dispositivo RAID (sólo elimina los metadatos del array, no el contenido de los discos):

mdadm --zero-superblock /dev/sdb /dev/sdc

Para simular el fallo de un disco:

mdadm --manage --set-faulty /dev/md0 /dev/sdc

Para sustituir un disco del RAID:

mdadm --remove /dev/md0 /dev/sdc

mdadm --add /dev/md0 /dev/sdd

Una vez creado el dispositivo es necesario establecer el sistema de archivos (mkfs) y montar la unidad (comando mount, archivo fstab...)

La configuración de los dispositivos RAID se guarda en el archivo /etc/mdadm.conf (no necesario a partir de RH6).

Para crear este archivo con los dispositivos RAID ya existentes:

mdadm --verbose --detail --scan > /etc/mdadm.conf

antiguo

T8: Almacenamiento

(Almacenamiento Local)

L M V

LVM / LVM2 → administración del espacio de Almacenamiento.

• Volumenes lógicos = Particiones físicas pero pueden extenderse sobre n.º de Almacenamiento.

Organización

- ↳ Volumenes físicos (1 por disco o partición)
- ↳ Grupos de volumenes (Agrupan volúmenes físicos)
- ↳ Volumenes lógicos (reciben power a la capacidad del grupo)

• Tb Pueden config volumenes físicos basados en RAID

Nota: Por simplicidad, la partición /boot no forma parte de los grupos de volúmenes

Comandos:

■ Si se va a usar un disco completo no debe tener ninguna partición.
Si se usa una partición, debe crearse de tipo LVM (opción t de fdisk, código 0x8e: Linux LVM)

■ Crear volúmenes físicos (pvcreate):

`pvcreate /dev/sda /dev/sdb /dev/sdf1`

■ Visualizar los volúmenes físicos. Distintos comandos:

`lvmiskscan`, `pvs`, `pvdisplay` y `pvscan`

■ Redimensionar un volumen físico (pvresize)

Se usa al cambiar el tamaño de una partición con fdisk

■ Eliminar un volumen físico:

`pvremove /dev/sd*`

■ Crear un grupo de volúmenes: `vgcreate vgl /dev/sdc /dev/sdd`

■ Visualizar los grupos de volúmenes:

`vgs`, `vgdisplay` y `vgscan`

■ Añadir volúmenes físicos a un grupo de volúmenes (`vgextend`): `vgextend vgl /dev/sdf1`

■ Eliminar un volumen físico del grupo de volúmenes (`vgreduce`): `vgreduce vgl /dev/sdd`

■ Eliminar un grupo de volúmenes (`vgremove`): `vgremove vgl`

■ Crear un volumen lógico (`lvcreate`):

`lvcreate -L 4G -n datos vgl`
`lvcreate -L 60VG -n arr vgl`
`lvcreate -L 100%FREE -n datos2 vgl`

■ Ver los volúmenes lógicos:

`lvs`, `lvdisplay` y `lvscan`

■ Redimensionar volúmenes lógicos (`lvreduce`, `lvextend`)

`lvreduce -size -4G /dev/vgl/datos`
`lvextend -L 12G /dev/vgl/datos`

■ Eliminar un volumen lógico (`lvremove`):

`lvremove /dev/vgl/datos`

Nota: será necesario extender el espacio de archivos redimensionado

Partición que queremos redimensionar

Almacenamiento en Red

- La unidad de almacenamiento (disco duro) NO ESTÁ conectada localmente
 - ↳ Puede mantenerse en Red de datos CONVENTIONAL o RED de almacenamiento. (SAN, Storage Area Network)
- 2 Niveles distintos de gestión
 - Nivel de clúster → F de protocolos: NFS, CIFS/SMB
 - Nivel de bloque → F de protocolos: FC / iSCSI
- 2 tipos de acceso
 - Exclusivo (recurso sólo de acceso a 1 cliente)
 - Compartido (pueden acceder varios clientes a la vez)

NFS (Network File System)

- Sist. de archivos distinto → permite al cliente Montar un directorio remoto que ha sido exportado por un servidor NFS
- Servidor sobre RPC [Remote Procedure Call]
 - ↳ Proprietary → review ejec. Procedim. en una máquina remota
 - ↳ Enlaza a protocolo TCP para tener q preocuparse por lo DCA entre ellos cuando es E/S
 - ↳ LEADUP, READ, WRITE, RECLAIM

Funcionamiento NFS

- Atajos NFS → completamente transparentes al usuario
 - ↳ Sys op local → maneja las peticiones de E/S sobre NFS
 - ↳ Todas las op E/S son idénticas y no dependen de si el acceso es local o remoto
 - Para ver los directorios exportados: `showmount -e dir_servidor`
 - Para montar la unidad NFS en el sistema de archivos: `mount -t nfs server:/var/pub /mnt/pub`
- (monta el directorio compartido /var/pub de la máquina server en el punto de montaje /mnt/pub)

Servicios RPC y Arranque

- ? Necesitamos 2 servicios RPC (mount, rpcbind) El modo de acceso es NFS (acceso a los archivos)
 - Para tener DCA RPC necesario conocer el puerto qnd ejecutan los servicios
 - ↳ Servicio mntbind + portmapper → escucha 111/udp capaz de mappear nombres del servicio → Puerto asignado
 - Para q sea info. ese dispositivo → servicios NFS deben registrarse en portMapper en su arranque (o rpcbind)
 - IMP. arrancar en el siguiente orden: 1º) rpcbind (o portMapper)
2º) nfs (servicio nfs-server)

Rocky: Paquete nfs-utils (cliente y servidor) y rpcbind. Servicio nfs-server.

Ubuntu: Paquetes nfs-kernel-server y nfs-common. Servicio nfs-kernel-server

NFS → Exportación de Archivos

- ? Necesitamos config. los directorios a exportar → las rutas desde los q se permite acceder
- ?/etc/exports

# Directory	Clientes(opciones)
/var/pub	172.25.11.0/24 (rw,all_squash)
/var/proyectos	pc01.efirel.com (rw) *.efirel.com ()

dirección o exportación

en cada línea debe indicarse → por ejemplo:

- + /var → modo Lectura // /var → Modo lectura/escritura
- + Share → la conexión NFS debe originarse en un puerto privilegiado (1-1023)
- + No-Selinux → check + uno que realizó el montaje → No verifica los directorios accedidos q están bajo él

+ root-Squash

↓ Imprime q root tiene privilegio

root@server:~\$ ls -l /tmp/archivo.txt
root@server:~\$ ls -l /tmp/nuevo_archivo.txt
root@server:~\$ rm -f /tmp/archivo.txt
rm: cannot remove '/tmp/archivo.txt': Permission denied
root@server:~\$ rm -f /tmp/nuevo_archivo.txt
rm: cannot remove '/tmp/nuevo_archivo.txt': Permission denied
No-root-Squash

+ all - squash / no-all - squash

Todos los Uso del cliente se convierte en el uso del usuario Américano

UID/GID en NFS

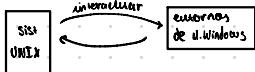
- Propiedad de un archivo se establece mediante UDO/EGO
 - ↳ Sigue/código local. Pero pueden almacenarse en Archivo
- Si vas a hacer un uso generalizado de los directorios controlado
 - Monitorizar en todos los bits los = UDO
 - Controlar cuentas de usuario en un solo SIST = NIS (revisor info. sys)
- C) Desnormalizar la propiedad de los nuevos directorios (figuras)

T8 : Almacenamiento

Almacenamiento en Red

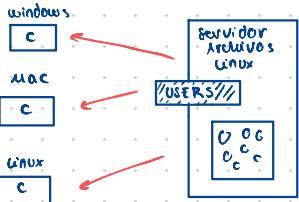
SMB / CIFS + Samba

- Samba → implementación libre del protocolo de archivos compartidos de Microsoft Windows (antes: SBS) → server Always Share → share → common → mount file sys
- Samba permite:



- + Resolución de nombres y exploración de Red (WINS & DNS)
- + Acceso a los recursos compartidos existentes
- + Validación local de usuarios
- + Gestión de validación en los dominios existentes
- + Exportación de directorios y recursos disponibles en clientes Windows

permite crear un servidor de Archivos Linux



Componentes Samba: demonios ↗ nmbd

- ↳ **nmbd** ↗ Resolución de nombres Anuncia grupos de trabajo → Reursos compartidos ↗ 139 + 445

- ↳ **smbd** ↗ Autentifica usuarios Ofrece recursos compartidos ↗ Archivos, impresoras ↗ 139/445 y 446/500

Rocky. Paquetes samba, samba-client y samba-common.

Servicios nmb y smb (levantar ambos).

Ubuntu: Paquetes samba y smbdclient. Servicios nmbd y smbd

Herramientas y utilidades SAMBA

- **findsmb.** Lista información de las máquinas
- **net.** Herramienta para administrar servidores Samba y CIFS remotos, similar a la utilidad net dos/Windows.
- **nmblookup.** Consulta nombres NetBIOS obteniendo su IP.
- **pdbedit.** Gestiona la base de datos SAM.
- **smblclient.** Utilidad de acceso SMB/CIFS, similar a FTP.
- **smbspasswd.** Cambia la contraseña Samba de un usuario.
- **smbsstatus.** Muestra las conexiones Samba actuales.
- **swat.** Herramienta de administración web para Samba.
- **smbfs.** Sistema de ficheros SMB para Linux. Permite montar recursos SMB de forma equivalente a NFS.
- **winbindd.** Cliente para resolver nombres en entornos NT.
- ...

Autenticación:

2 Modos

- + **share** (obsoleto) → Protege cada recurso → Contraseña
 - + **user** → Serv → recibe credenciales usuario de acceso Clientes → Local → A través de uno controlador (POC/POC o AD)
- ↳ una vez autentif. → se contrastan sus derechos sobre el recurso accedido

Config Samba (smb.conf) → ? /etc/Samba

3 secciones:

- [global] → Param globales [revisor] → Param defecto Punto de sus servicios (niveles 0 por ejemplo) share, user, domain o ad
- [homes] → Define Recurso de Red Para cada usuario Samba asociado a su home
- [printers] → Define Recurso de Red Para cada impresora reconocida por samba

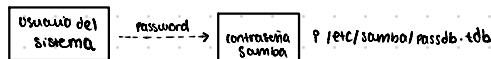
los recursos compartidos tienen propiedades asociadas:

[MiCarpet]

Path = /directorio	Directorio o cola de impresora a compartir
Comment = Comentario	Descripción del recurso
Browsable = yes no	Visibilidad del recurso
Writable = yes no	Derechos de escritura
Public = yes no	Recurso público
Valid users = usuarios	Usuarios autorizados
Create mask = máscara	Máscara para nuevos archivos

usuarios de Samba

- Si se usa autentif. local → necesario crear cuentas Samba



SMBpasswd → a ejecución → asocia usuario al de Samba
se puede modificar a posteriori
guardando → a

iSCSI

- **internat** Small Computer System Interface → SCSI → interfaz de acceso a unidades de almacenamiento local (ATA/SATA, NVMe...)

- Implementación del estándar iSCSI sobre una red IP
 - ↳ Alternativa → Fibre Channel

- Opera a nivel de Bloque:

- ↳ **iSCSI target node** (servidor → target) → se conecta → **iSCSI Initiator (cliente)**

- Ofrece espacio de Almacenamiento organizado en volúmenes → LVM
- Incluye los driver de acceso + programa para gestionar conexiones,

- **IQN (iSCSI Qualified Name)** : identif. único para un elem iSCSI
- Alternativa → **EUI (Extended Unique Identifier)**

T8 : Almacenamiento

Almacenamiento en Red

iSCSI

Configurar un iSCSI Target

» targetcli ~> shell nro < root admin recursos target

- Cada Recurso o unidad logica (LUN)

lun por defecto

un bloque de Almacenamiento --> se exporta vía iSCSI

- 3 fuentes para generar unidades logicas Target (Backstore)

disks, ramdisks, partitions, volúmenes

- una Unidad (LUN) se identifica WWN (worldWide Name)
(normalmente autogenerado y con formato ION, EUI64 NAN)

Proceso a seguir:

- 1) Crear Backstore (blocks)
- 2) Establecer autenticación para el acceso
- 3) Asignar un WWNN y un Portal (dirección, IP+puertos de escucha)
- 4) Crear un LUN
- 5) Establecer la ACL para un acceso remoto

instalamos el paquete targetcli e iniciamos la utilidad

[root@server ~]# yum install -y targetcli

[root@server ~]# systemctl enable targetcli

[root@server ~]# systemctl start targetcli

[*] Comemos un backstore

/dev

o- /backstores

o- block [Storage Objects: 0]

o- fileio [Storage Objects: 0]

o- pascal [Storage Objects: 0]

o- ramdisk [Storage Objects: 0]

o- iscsi [Targets: 0]

o- luns [Targets: 0]

o- block/luns [Storage Objects: 0]

/backstores/block/ create vol01 /dev/vgl/datos

Created block storage object vol01 using /dev/vgl/datos.

/backstores/block/ ls

o- block [Storage Objects: 1]

o- vol01 [/dev/vgl/datos (26.4GB) write-thru deactivated]

Creamos un LUN asociado al backstore:

[root@server ~]# iosciladm -cd iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d/

/iscsi/iqn.2003-01.org.linux-

/iscsi/iqn.2003-01.org.linux- create /backstores/block/vol01

Created LUN 1 using /backstores/block/vol01

[root@server ~]# iosciladm -cd iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d/

/iscsi/iqn.2003-01.org.linux- cd /portals/

/iscsi/iqn.2003-01.org.linux- create

Using default IP port 3260

Binding to INADDR_ANY (0.0.0.0:3260).

Created network portal 0.0.0.0:3260.

Configurar un iSCSI Initiator

» iSCSIADM: utilidad para gestionar las conexiones a recursos iSCSI

- Mecanismo autenticación + credenciales → /etc/iscsi/iscsid.conf
- Unidades iSCSI invitadas → ? /etc/iscsi/initiatorname.iscsi
- Proceso para iniciar un recurso iSCSI:
 - Obtener los nombres iqn (LUN) exportados por el SERVIDOR_ISCSI
 - Config nombre iqn del recurso (? initiatorname.iscsi)
 - (Config autenticación ? iscsid.conf)

instalamos el paquete iscsi-initiator-utils

[root@server ~]# yum install -y iscsi-initiator-utils

Obtenemos los iqn que exporta el servidor

192.168.211.138:3260,1 iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d

Comprobamos el estatus del target obtenido

[root@server ~]# iscsidadm -m node -o show

node_name: iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d

node.tgtpt = 1

node.startup = automatic

...

Añadimos el iqn en initiatorname.iscsi

[root@server ~]# vi /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d

Establecemos la autenticación (no necesaria en esta demo)

[root@server ~]# vi /etc/iscsi/iscsid.conf

Conectamos al recurso iSCSI

[root@server ~]# iscsidadm -m node -o login

Login in to [iface: default, target: iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d, portal: 192.168.211.138,3260]

(multiple)

Login to [iface: default, target: iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d, portal: 192.168.211.138,3260]

successful.

Verificamos la conexión establecida

[root@server ~]# iscsidadm -m session -o show

tcp: [3] 192.168.211.138:3260,1 iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d (non-flash)

Comprobamos que se ha añadido una nueva partición (sdb)

[root@server ~]# cat /proc/partitions -- isblk

major minor #blocks name

8	0	20971520	sda
8	1	512000	sdal
8	2	2048000	sda2
11	0	1048576	srd0
253	0	1073741824	dm-0
253	1	2097152	dm-1
8	16	10485760	sdb

Trabajamos con sdb como si se tratara de una unidad local

...

Autenticación: en este ejemplo no activamos auth
/backstores/block/ cd ./, /iscsi
/iscsi> set discovery_auth enable=0
Para tener que configurar el servidor para su arranque automático

/iscsi> create

Created target iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d.

Created TPG1

Global pref auto_add_default_portal=true

Created default portal listening on all IPs (0.0.0.0), port 3260.

/iscsi> ls

o- iscsi [Targets: 1]

o- iqn.2003-01.org.linux-iscsi.server.x8664.sn.d06f64e0d23d

[TPG1: 1]

o- portal1 [Hosts: 1, Initiators: 1, LUNS: 0]

o- srd0 [LUNS: 0]

o- luns [LUNS: 0]

o- portals [Portals: 1]

o- 0.0.0.0:3260 [OK]

No establecemos ninguna restricción de acceso (solo para la demo)

/iscsi> cd iqn.2003-01.org.linux-

iscsi.server.x8664.sn.d06f64e0d23d/

/iscsi/iqn.2003-01.org.linux- cd tpg1/luns

/iscsi/iqn.2003-01.org.linux- create /backstores/block/vol01

Created LUN 1 using /backstores/block/vol01

/iscsi> ls

o- block [Storage Objects: 1]

o- luns [Block/vol01:1 /dev/vgl/datos]

Creamos un portal (Nota: puede que ya esté creado)

/iscsi/iqn.2003-01.org.linux- cd /portals/

/iscsi/iqn.2003-01.org.linux- create

Using default IP port 3260

Binding to INADDR_ANY (0.0.0.0:3260).

Created network portal 0.0.0.0:3260.

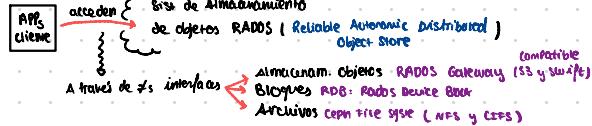
[root@server ~]# cat /etc/iscsi/saveconfig.json

T8 : Almacenamiento

Clúster de Almacenamiento

ceph

- Infra. de Almacenam. distribuido de objetos → Cluster escalable Alta disponib.
- ↳ Basado en sus + Abierto (GPL) + adaptable (no requiere nu. expens.)

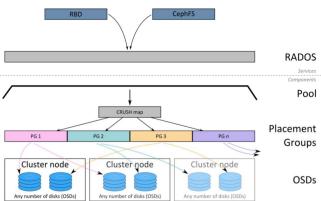


Componentes fundamentales

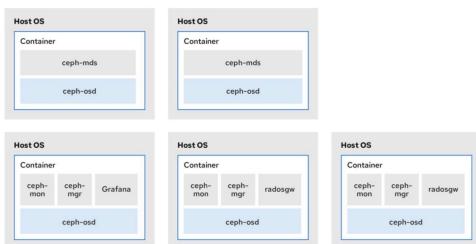
- ↳ OSD (Object Store Daemons): gestionan interfaz con discos /fisicos
- ↳ MON (Monitors): mantienen un mapa con la estructura del cluster
- ↳ MDS (Meta Data Server): mantiene las metadatos de los objetos
- ↳ RGW (RADOS Gateway): interfaz de obj compatible con S3 y Swift

CRUSH

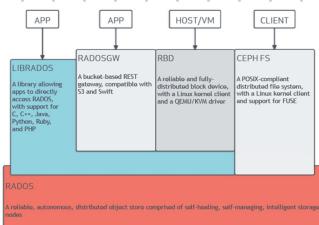
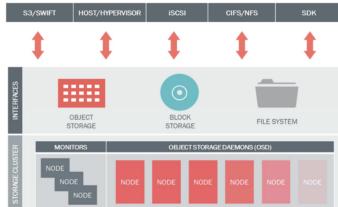
Controlled Replication Under Scalable hashing
algoritmo q det. cómo se almacenan los obj



Ejemplo de infraestruct. en ceph



Interfaces y Componentes:



Componentes del sistema (demonios)

- **ceph-mon** (Monitor). Mantiene una copia del mapa de almacenamiento. Debe instalarse en al menos un nodo.
- **ceph-mgr** (Manager). Interface de monitoreo y gestión. Normalmente se instala en todos los nodos monitores.
- **ceph-osd** (Object Storage Device). Son los nodos de almacenamiento. Se requieren al menos tres.
- **ceph-mds** (Meta Data Server). Almacena los metadatos y coordina el acceso al almacenamiento compartido.
- **ceph-radosgw** (Object Gateway). Ofrece interfaces para el acceso a los objetos (S3, Swift).

