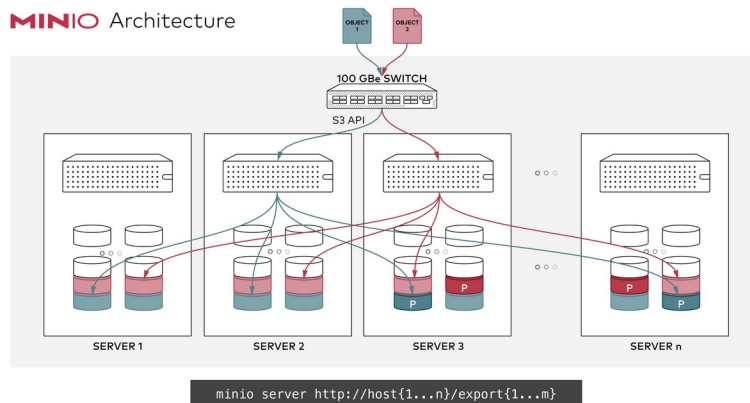




Se plantea un servicio de almacenamiento simple 'on-premise' y virtualizado en RedHat KVM para 4 nodos y 4 volúmenes: FS -> VDO|xfs. En modo distribuido.



Virtualización rhel → KVM

<<<<<<< host's distribuidos >>>>>>>

192.168.10.151	rhel01.cadilinea.lan	rhel01
192.168.10.152	rhel02.cadilinea.lan	rhel02
192.168.10.153	rhel03.cadilinea.lan	rhel03
192.168.10.154	rhel04.cadilinea.lan	rhel04

→ Preparación para el poblado de: Discos/Isos/Redes → Ambiente: → --os-variant → rhel8.3

Se elige una red para el laboratorio personalizada, y con el objetivo de entrenamiento.

```
redhat00 ~ # mkdir /var/lib/libvirt/Discos -p
redhat00 ~ # mkdir /var/lib/libvirt/Isos -p
redhat00 ~ # mkdir /var/lib/libvirt/Redes -p
```

→ Creación del: bridge → br10 en modo NAT.

```
redhat00 ~ # virsh net-list --all
Nombre Estado Inicio automático Persistente
-----
default activo si si
```

Creamos un 'bridge' personalizado → 'br10'

```
redhat00 ~ # vim /var/lib/libvirt/Redes/br10.xml
<network>
```



```
<name>br10</name>
<forward mode='nat'>
  <nat>
    <port start='1024' end='65535'/>
  </nat>
</forward>
<bridge name='br10' stp='on' delay='0'/>
<ip address='192.168.10.5' netmask='255.255.255.0'>
  <dhcp>
    <range start='192.168.10.151' end='192.168.10.165'/>
  </dhcp>
</ip>
</network>
```

redhat00 ~ # virsh net-define /var/lib/libvirt/Redes/br10.xml

La red br10 se encuentra definida desde /var/lib/libvirt/Redes/br10.xml

redhat00 ~ # virsh net-list --all

Nombre	Estado	Inicio automático	Persistente
br10	inactivo	no	si
default	inactivo	si	si

redhat00 ~ # virsh net-start br10

La red br10 se ha iniciado

redhat00 ~ # virsh net-autostart br10

La red br10 ha sido marcada para iniciarse automáticamente

redhat00 ~ # virsh net-autostart default --disable

Ha sido quitada la marca de la red default para que pueda iniciarse automáticamente

redhat00 ~ # virsh net-list --all

Nombre	Estado	Inicio automático	Persistente
br10	activo	si	si
default	inactivo	no	si

redhat00 ~ # ip addr show dev br10

```
6: br10: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default qlen 1000
    link/ether 52:54:00:db:ea:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.5/24 brd 192.168.10.255 scope global br10
        valid_lft forever preferred_lft forever
```

→ **Definimos la Configuración de las distintas máquinas virtuales:**

redhat00 ~ # osinfo-query os | grep -i rhel8

```
rhel8.0          | Red Hat Enterprise Linux 8.0          | 8.0          | http://redhat.com/rhel/8.0
```



rhel8.1	Red Hat Enterprise Linux 8.1	8.1	http://redhat.com/rhel/8.1
rhel8.2	Red Hat Enterprise Linux 8.2	8.2	http://redhat.com/rhel/8.2
rhel8.3	Red Hat Enterprise Linux 8.3	8.3	http://redhat.com/rhel/8.3

→ [Descargamos la ISO a instalar previamente en la ruta: /var/lib/libvirt/Isos/](#)

```
redhat00 ~ # ls /var/lib/libvirt/Isos/  
rhel-8.3-x86_64-dvd.iso
```

→ **Instalamos** → **rhel01** → [192.168.10.151](#)

```
redhat00 ~ # virt-install \  
--name rhel01 \  
--ram 3072 \  
--vcpus 2 \  
--disk path=/var/lib/libvirt/Discos/disco1-rhel01-25GB,size=25,bus=virtio,format=qcow2 \  
--disk path=/var/lib/libvirt/Discos/disco2-rhel01-20GB,size=20,bus=virtio,format=qcow2 \  
--os-variant rhel8.3 \  
--os-type linux \  
--network bridge=br10 \  
--graphics vnc \  
--console pty,target_type=serial \  
--location /var/lib/libvirt/Isos/rhel-8.3-x86_64-dvd.iso \  
--extra-args 'console=ttyS0,115200n8 serial'
```

```
Empezando la instalación...  
Obteniendo archivo vmlinuz... | 9.1 MB  
00:00:00  
Obteniendo archivo initrd.img... | 70 MB  
00:00:00  
Asignando 'disco1-rhel01-25GB' | 25 GB  
00:00:02  
Asignando 'disco2-rhel01-20GB' | 20 GB  
00:00:03  
Domain installation still in progress.  
Waiting for installation to complete.  
Running graphical console command: virt-viewer --connect qemu:///system --wait rhel01  
<ctrl>c
```

```
redhat00 ~ # virsh console --domain rhel01
```

```
Conectado con el dominio rhel01  
Carácter de escape es ^]  
Please make a selection from the above ['b' to begin installation, 'q' to quit,  
'r' to refresh]: r  
Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to  
refresh]: c
```

=====
Installation

- | | |
|---|--|
| 1) <input checked="" type="checkbox"/> Language settings
(Spanish (Spain)) | 2) <input checked="" type="checkbox"/> Time settings
(Europe/Madrid timezone) |
| 3) <input checked="" type="checkbox"/> Installation source
(Local media) | 4) <input checked="" type="checkbox"/> Software selection
(Minimal Install) |
| 5) <input checked="" type="checkbox"/> Installation Destination
(Automatic partitioning
selected) | 6) <input checked="" type="checkbox"/> Kdump
(Kdump is enabled) |
| 7) <input type="checkbox"/> Network configuration
(Not connected) | 8) <input checked="" type="checkbox"/> Root password
(Password is set.) |
| 9) <input checked="" type="checkbox"/> User creation
(User labs will be created) | |

Please make a selection from the above ['b' to begin installation, 'q' to quit, 'r' to refresh]: **b**

Progress

```
.
Setting up the installation environment
Setting up org_fedora_oscaped addon
Setting up com_redhat_kdump addon
..
Configuring storage
Creating disklabel on /dev/vda
Creating xfs on /dev/vda1
Creating lvmpv on /dev/vda2
Creating swap on /dev/mapper/rhel-swap
Creating xfs on /dev/mapper/rhel-root
...
Running pre-installation scripts
.
Running pre-installation tasks
...
Installing.
Starting package installation process
Downloading packages
Preparing transaction from installation source
Installing libgcc.x86_64 (1/602)
Installing hwdata.noarch (2/602)
Installing bind-license.noarch (3/602)
Installing fontpackages-filesystem.noarch (4/602)
Installing geolite2-country.noarch (5/602)
Installing geolite2-city.noarch (6/602)
Installing tzdata.noarch (7/602)
```



```
Installing python3-setuptools-wheel.noarch (8/602)
Installing python3-pip-wheel.noarch (9/602)
Installing libreport-fsfilesystem.x86_64 (10/602)
Installing dnf-data.noarch (11/602)
Installing dejavu-fonts-common.noarch (12/602)
Installing dejavu-sans-mono-fonts.noarch (13/602)
Installing abattis-cantarell-fonts.noarch (14/602)
Installing xkeyboard-config.noarch (15/602)
Installing vim-filesystem.noarch (16/602)
Installing tigervnc-license.noarch (17/602)
```

...

Storing configuration files and kickstarts

.

Installation complete

Use of this product is subject to the license agreement found at:
[/usr/share/redhat-release/EULA](#)

Installation complete. Press ENTER to quit:

...

```
[ OK ] Stopped Create System Users.
[ OK ] Stopped Remount Root and Kernel File Systems.
[ OK ] Reached target Shutdown.
[ OK ] Reached target Final Step.
      Starting Reboot...
[ OK ] Stopped Device-Mapper Multipath Device Controller.
dracut Warning: Killing all remaining processes
Rebooting.
[ 1904.450541] reboot: Restarting system
```

```
redhat00 ~ # virsh list --all
```

```
Id Nombre Estado
```

```
-----
```

```
- rhel01 apagado
```

```
redhat00 ~ # virsh start --domain rhel01
```

```
Se ha iniciado el dominio rhel01
```

```
redhat00 ~ # virsh list --all
```

```
Id Nombre Estado
```

```
-----
```

```
3 rhel01 ejecutando
```

```
redhat00 ~ # virsh console --domain rhel01
```

```
Conectado con el dominio rhel01
```

```
Carácter de escape es ^]
```



Red Hat Enterprise Linux 8.3 (Ootpa)
Kernel 4.18.0-240.el8.x86_64 on an x86_64

Activate the web console with: `systemctl enable --now cockpit.socket`

rhel01 login: root

Contraseña:

rhel01 ~ # dnf update

Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.

Error: No hay repositorios habilitados en "/etc/yum.repos.d", "/etc/yum/repos.d",
"/etc/distro.repos.d".

////////////////////////////////////
Es necesario disponer de una suscripción RedHat. No obstante todo deberá ser compatible para Oracle Linux/Fedora/CentOS.
////////////////////////////////////

rhel01 ~ # subscription-manager register --username xxxxxx --password xxxxx

Registrándose a: subscription.rhsm.redhat.com:443/subscription

El sistema ha sido registrado con ID: 7dfef3c5-7361-40bc-b69c-5d12fjcf4d40

El nombre del sistema registrado es: rhel01.cadilinea.lan

→ **Instalar MinIO de forma Permanente y como Servicio** → [/minio/data_01](#)

rhel01 ~ # cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

Virtualización rhel.

192.168.10.151	rhel01.cadilinea.lan	rhel01
192.168.10.152	rhel02.cadilinea.lan	rhel02
192.168.10.153	rhel03.cadilinea.lan	rhel03
192.168.10.154	rhel04.cadilinea.lan	rhel04

rhel01 ~ # useradd -r -m -d /opt/minio minio

rhel01 ~ # mkdir -p /opt/minio/bin

rhel01 ~ # chown -R minio:minio /opt/minio/

rhel01 ~ # wget -c https://dl.minio.io/server/minio/release/linux-amd64/minio -O /opt/minio/bin/minio

rhel01 ~ # chmod +x /opt/minio/bin/minio

rhel01 ~ # cp /opt/minio/bin/minio /usr/bin/minio



```
rhel01 ~ # chown minio:minio /opt/minio/bin/minio
```

```
rhel01 ~ # mkdir /minio/data_01 -p
```

```
rhel01 ~ # chown -R minio:minio /minio/data_01
```

```
rhel01 ~ # vim /etc/default/minio
```

```
# Volume to be used for MinIO server.
```

```
MINIO_VOLUMES="/minio/data_01/"
```

```
# Use if you want to run MinIO on a custom port.
```

```
MINIO_OPTS="--address :9000"
```

```
# Root user for the server.
```

```
MINIO_ROOT_USER=admin
```

```
# Root secret for the server.
```

```
MINIO_ROOT_PASSWORD=Admin#52
```

```
rhel01 ~ # vim /etc/systemd/system/minio.service
```

```
[Unit]
```

```
Description=MinIO
```

```
Documentation=https://docs.min.io
```

```
Wants=network-online.target
```

```
After=network-online.target
```

```
AssertFileIsExecutable=/usr/bin/minio
```

```
[Service]
```

```
WorkingDirectory=/opt/minio/
```

```
User=minio
```

```
Group=minio
```

```
EnvironmentFile=/etc/default/minio
```

```
ExecStartPre=/bin/bash -c "if [ -z \"${MINIO_VOLUMES}\" ]; then echo \"Variable MINIO_VOLUMES not set in /etc/default/minio\"; exit 1; fi"
```

```
ExecStart=/usr/bin/minio server http://rhel0{1...4}/minio/data_01
```

```
# Let systemd restart this service always
```

```
Restart=always
```

```
# Specifies the maximum file descriptor number that can be opened by this process
```

```
LimitNOFILE=65536
```

```
# Specifies the maximum number of threads this process can create
```

```
TasksMax=infinity
```

```
# Disable timeout logic and wait until process is stopped
```

```
TimeoutStopSec=infinity
```

```
SendSIGKILL=no
```



[Install]

WantedBy=multi-user.target

rhel01 ~ # systemctl daemon-reload

rhel01 ~ # firewall-cmd --permanent --add-port=9000/tcp

rhel01 ~ # firewall-cmd --reload

redhat00 ~ # virsh shutdown --domain rhel01

redhat00 ~ # virsh snapshot-create-as --domain rhel01 --name "Master-rhel01" --atomic

→ Clonación de Instancias → **rhel02, rhel03 y rhel04**

redhat00 ~ # virt-clone --original rhel01 --name rhel02 --auto-clone

Asignando 'disco1-rhel01-25GB-clone' | 25 GB 00:01:31

Asignando 'disco2-rhel01-20GB-clone' | 20 GB 00:00:00

El clon 'rhel02' ha sido creado exitosamente.

redhat00 ~ # virsh list --all

Id Nombre Estado

```
-----  
- rhel01 apagado  
- rhel02 apagado
```

redhat00 ~ # tree /var/lib/libvirt/Discos/

```
/var/lib/libvirt/Discos/  
├── disco1-rhel01-25GB  
├── disco1-rhel01-25GB-clone  
├── disco2-rhel01-20GB  
└── disco2-rhel01-20GB-clone
```

redhat00 ~ # cd /var/lib/libvirt/Discos/

redhat00 Discos # mv disco1-rhel01-25GB-clone disco1-rhel02-25GB

redhat00 Discos # mv disco2-rhel01-20GB-clone disco2-rhel02-20GB

redhat00 Discos # tree /var/lib/libvirt/Discos/

```
/var/lib/libvirt/Discos/  
├── disco1-rhel01-25GB  
├── disco1-rhel02-25GB  
├── disco2-rhel01-20GB  
└── disco2-rhel02-20GB
```

redhat00 Discos # cd

redhat00 ~ # virsh edit --domain rhel02

...

<devices>



```

<emulator>/usr/libexec/qemu-kvm</emulator>
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='/var/lib/libvirt/Discos/disco1-rhel02-25GB'/>
  <target dev='vda' bus='virtio'/>
  <address type='pci' domain='0x0000' bus='0x04' slot='0x00' function='0x0'/>
</disk>
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='/var/lib/libvirt/Discos/disco2-rhel02-20GB'/>
  <target dev='vdb' bus='virtio'/>
  <address type='pci' domain='0x0000' bus='0x05' slot='0x00' function='0x0'/>
</disk>

```

...

```
redhat00 ~ # virsh start --domain rhel02
```

```
redhat00 ~ # ssh 192.168.10.151
```

```
rhel01 ~ # hostnamectl set-hostname rhel02.cadilinea.lan
```

```
rhel01 ~ # su -
```

```
rhel02 ~ # ip a show enp1s0 | grep inet
```

```
inet 192.168.10.151/24 brd 192.168.10.255 scope global noprefixroute enp1s0
```

```
inet6 fe80::a4e6:cd7b:f18b:b030/64 scope link noprefixroute
```

```
rhel02 ~ # nmcli connection modify enp1s0 ipv4.addresses 192.168.10.152
```

```
rhel02 ~ # shutdown -h 0
```

==> [Repetimos el procedimiento para rhel03 y rhel04, ...](#)

////////////////////////////////////
Es importante asignar los hosts e IP's adecuados, y según el esquema previsto.
 //////////////////////////////////////

```
redhat00 ~ # tree /var/lib/libvirt/Discos/
```

```
/var/lib/libvirt/Discos/
```

```
├── disco1-rhel01-25GB
├── disco1-rhel02-25GB
├── disco1-rhel03-25GB
├── disco1-rhel04-25GB
├── disco2-rhel01-20GB
├── disco2-rhel02-20GB
├── disco2-rhel03-20GB
└── disco2-rhel04-20GB
```

→ La Configuración actual deberá ser la siguiente:

```
## Virtualización rhel.
```

```
192.168.10.151          rhel01.cadilinea.lan  rhel01
```

```
192.168.10.152          rhel02.cadilinea.lan  rhel02
```

```
192.168.10.153          rhel03.cadilinea.lan  rhel03
```



```
192.168.10.154          rhel04.cadilinea.lan  rhel04
```

```
redhat00 ~ # for i in {1..4}; do virsh snapshot-create-as --domain rhel0$i --name "Inicial" --atomic ; done
```

```
redhat00 ~ # for i in {1..4}; do virsh snapshot-list --domain rhel0$i ; done
```

```
Nombre      Hora de creación      Estado
-----
Master-rhel01 2021-02-22 07:57:04 +0100  shutoff
Inicial      2021-02-22 09:02:11 +0100  shutoff
```

```
Nombre      Hora de creación      Estado
-----
Inicial 2021-02-22 09:02:12 +0100  shutoff
```

```
Nombre      Hora de creación      Estado
-----
Inicial 2021-02-22 09:02:13 +0100  shutoff
```

```
Nombre      Hora de creación      Estado
-----
Inicial 2021-02-22 09:02:13 +0100  shutoff
```

→ [Iniciamos la configuración para los volúmenes distribuidos de MinIO:](#)

```
////////////////////////////////////
Se pudo haber hecho la configuración subsiguiente y haber clonado con anterioridad. No utilizo
los UUID habida cuenta que podrían clonar el mismo hash. En la práctica no ocurre. Lo
complico a propósito para evitarlo. Los sistemas clonados no los incorporo en /etc/fstab con su
UUID. Como son sistemas clonados los monto con su /dev/mapper.
////////////////////////////////////
```

```
redhat00 ~ # for i in {1..4}; do virsh start --domain rhel0$i ; done
```

```
Se ha iniciado el dominio rhel01
Se ha iniciado el dominio rhel02
Se ha iniciado el dominio rhel03
Se ha iniciado el dominio rhel04
```

```
redhat00 ~ # ssh rhel01
```

```
rhel01 ~ # lsblk
```

```
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0            11:0  1 1024M  0 rom
vda            252:0  0   25G  0 disk
├─vda1         252:1  0    1G  0 part /boot
└─vda2         252:2  0   24G  0 part
   └─rhel-pool00_tmeta 253:0  0   12M  0 lvm
      └─rhel-pool00-tpool 253:2  0  16,7G  0 lvm
         └─rhel-root    253:3  0  16,7G  0 lvm /
            └─rhel-pool00 253:5  0  16,7G  1 lvm
```



```

├─rhel-pool00_tdata 253:1 0 16,7G 0 lvm
│ └─rhel-pool00-tpool 253:2 0 16,7G 0 lvm
│   └─rhel-root 253:3 0 16,7G 0 lvm /
│     └─rhel-pool00 253:5 0 16,7G 1 lvm
└─rhel-swap 253:4 0 2,5G 0 lvm [SWAP]
vdb 252:16 0 20G 0 disk

```

-> **SOLO: Si hubiese datos previos en el volúmen 'vdx' → Recomendación de Intel para SSD.**

////////////////////////////////////
No es muy recomendable hacer los 'zeroes' y según recomienda Intel y desde mi punto de vista. Pero es Intel quien lo recomienda.
 //////////////////////////////////////

```

rhel01 ~ # nohup dd if=/dev/zero of=/dev/vdb oflag=direct bs=2M &
[1] 1662
rhel01 ~ # nohup: se descarta la entrada y se añade la salida a 'nohup.out'
<Intro>

```

→ **Seguimos aquí ...**

////////////////////////////////////
Utilizo VDO 'puro' para crear el FS. Como recomendación importante puede utilizarse en combinación con LVM también. Los vg's y lv's podrán ampliarse mas fácilmente. Sinceramente no lo veo claro. Y menos con la opción 'discard' en el montaje del FS.
 //////////////////////////////////////

```

rhel01 ~ # vdo create --name minio-vol01 --device /dev/vdb --vdoSlabSize 128
Creating VDO minio-vol01
  Logical blocks defaulted to 4484120 blocks.
  The VDO volume can address 17 GB in 138 data slabs, each 128 MB.
  It can grow to address at most 1 TB of physical storage in 8192 slabs.
  If a larger maximum size might be needed, use bigger slabs.
Starting VDO minio-vol01
Starting compression on VDO minio-vol01
VDO instance 0 volume is ready at /dev/mapper/minio-vol01

```

```

rhel01 ~ # lsblk /dev/vdb -f
NAME      FSTYPE LABEL UUID                                MOUNTPOINT
vdb       vdo     450ea16a-212b-4182-a75b-7f84c700d4a3
└─minio-vol01

```

```

rhel01 ~ # vdo stats --human-readable
Device          Size  Used Available Use% Space saving%
/dev/mapper/minio-vol01 20.0G  2.9G  17.1G  14%      N/A

```

```

rhel01 ~ # vdo status --name minio-vol01 | less

```



VDO status:

Date: '2021-02-22 09:24:12+01:00'

Node: rhel01.cadilinea.lan

Kernel module:

Loaded: true

Name: kvdo

Version information:

kvdo version: 6.2.3.114

Configuration:

File: /etc/vdoconf.yml

Last modified: '2021-02-22 09:19:20'

VDOs:

minio-vol01:

Acknowledgement threads: 1

Activate: enabled

Bio rotation interval: 64

Bio submission threads: 4

Block map cache size: 128M

Block map period: 16380

Block size: 4096

CPU-work threads: 2

Compression: enabled

Configured write policy: auto

Deduplication: enabled

Device mapper status: 0 35872960 vdo /dev/vdb normal - online online 753050 5242880

...

```
rhel01 ~ # vdo status --name minio-vol01 |grep -i compression
```

Compression: enabled

```
rhel01 ~ # vdo status --name minio-vol01 |grep -i deduplication
```

Deduplication: enabled

```
rhel01 ~ # mkfs.xfs /dev/mapper/minio-vol01
```

meta-data=/dev/mapper/minio-vol01 isize=512 agcount=4, agsize=1121030 blks

= sectsz=4096 attr=2, projid32bit=1

= crc=1 finobt=1, sparse=1, rmapbt=0

= reflink=1

data = bsize=4096 blocks=4484120, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0, ftype=1

log =internal log bsize=4096 blocks=2560, version=2

= sectsz=4096 sunit=1 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

Discarding blocks...Done.

```
rhel01 ~ # lsblk -f /dev/vdb
```

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
------	--------	-------	------	------------



```
vdb      vdo      450ea16a-212b-4182-a75b-7f84c700d4a3
└─minio-vol01 xfs      e3370d5c-b441-42d0-a6d1-86ad73ee13d2
```

////////////////////////////////////
*Si el sistema **NO** es Clonado, deberá utilizarse la UUID del dispositivo. NO un /dev/mapper, ...
En la práctica funcionará igual. Pero la 'heal' no será la misma.*
////////////////////////////////////

```
rhel01 ~ # vim /etc/fstab
```

```
...
```

```
## Montaje para MinIO
```

```
/dev/mapper/minio-vol01 /minio/data_01 xfs defaults 0 0
```

```
rhel01 ~ # mount -a
```

```
rhel01 ~ # lsblk -f /dev/vdb
```

```
NAME      FSTYPE LABEL UUID                                MOUNTPOINT
vdb      vdo      450ea16a-212b-4182-a75b-7f84c700d4a3
└─minio-vol01 xfs      e3370d5c-b441-42d0-a6d1-86ad73ee13d2 /minio/data_01
```

```
rhel01 ~ # chown minio:minio -R /minio/data_01/
```

```
rhel01 ~ # chmod u+rwx -R /minio/data_01/
```

```
rhel01 ~ # vim .bashrc
```

```
...
```

```
export MINIO_ROOT_USER=admin
```

```
export MINIO_ROOT_PASSWORD=Admin#52
```

////////////////////////////////////
Puede utilizarse un método menos transparente para exportar las variables de entorno y que son de acceso, ...
////////////////////////////////////

```
rhel01 ~ # source .bashrc
```

```
rhel01 ~ # vim /etc/systemd/system/minio.service
```

```
...
```

```
ExecStart=/usr/bin/minio server http://rhel0{1...4}/minio/data_01
```

→ Repetimos para: **rhel02**, **rhel03**, y **rhel04**, ...

```
redhat00 ~ # for i in {1..4}; do virsh snapshot-create-as --domain rhel0$i --name "minio-vol01" --atomic ; done
```

→ Comprobamos si todo OK para el servicio → **minio.service**

```
redhat00 ~ # for i in {1..4}; do virsh start --domain rhel0$i ; done
```

```
redhat00 ~ # virsh list --all
```

```
Id Nombre Estado
```



```
-----
65 rhel01 ejecutando
66 rhel02 ejecutando
67 rhel03 ejecutando
68 rhel04 ejecutando
```

```
redhat00 ~ # ssh rhel01
```

```
rhel01 ~ # systemctl enable --now minio.service
```

```
rhel01 ~ # systemctl status minio.service
```

```
● minio.service - MinIO
```

```
Loaded: loaded (/etc/systemd/system/minio.service; disabled; vendor preset: disabled)
```

```
Active: active (running) since Mon 2021-02-22 17:19:43 CET; 7min ago
```

```
Docs: https://docs.min.io
```

```
Process: 19975 ExecStartPre=/bin/bash -c if [ -z "${MINIO_VOLUMES}" ]; then echo "Variable MINIO_VOLUMES not set in /etc/default/mini>
```

```
Main PID: 19977 (minio)
```

```
Tasks: 8
```

```
Memory: 163.8M
```

```
CGroup: /system.slice/minio.service
```

```
└─19977 /usr/bin/minio server --address :9000 /minio/data_01/
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Attempting encryption of all config, IAM users and policies on MinIO backend
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Endpoint: http://192.168.10.151:9000
```

```
http://127.0.0.1:9000
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Browser Access:
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: http://192.168.10.151:9000
```

```
http://127.0.0.1:9000
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Object API (Amazon S3 compatible):
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Go: https://docs.min.io/docs/golang-client-quickstart-guide
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Java: https://docs.min.io/docs/java-client-quickstart-guide
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: Python: https://docs.min.io/docs/python-client-quickstart-guide
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: JavaScript:
```

```
https://docs.min.io/docs/javascript-client-quickstart-guide
```

```
feb 22 17:19:44 rhel01.cadilinea.lan minio[19977]: .NET: https://docs.min.io/docs/dotnet-client-quickstart-guide
```

```
redhat00 ~ # for i in {1..4}; do virsh snapshot-create-as --domain rhel0$i --name "minio.service" --atomic ; done
```

```
redhat00 ~ # for i in {1..4}; do virsh snapshot-list --domain rhel0$i --topological; done
```

```
Nombre          Hora de creación      Estado
```

```
-----
Master-rhel01   2021-02-22 07:57:04 +0100  shutoff
```



```
Inicial      2021-02-22 09:02:11 +0100  shutoff
minio-vol01  2021-02-22 10:26:12 +0100  shutoff
minio.service 2021-02-22 18:16:23 +0100  running
```

```
Nombre      Hora de creación      Estado
-----
```

```
Inicial      2021-02-22 09:02:12 +0100  shutoff
minio-vol01  2021-02-22 10:26:13 +0100  shutoff
minio.service 2021-02-22 18:16:54 +0100  running
```

```
Nombre      Hora de creación      Estado
-----
```

```
Inicial      2021-02-22 09:02:13 +0100  shutoff
minio-vol01  2021-02-22 10:26:13 +0100  shutoff
minio.service 2021-02-22 18:17:15 +0100  running
```

```
Nombre      Hora de creación      Estado
-----
```

```
Inicial      2021-02-22 09:02:13 +0100  shutoff
minio-vol01  2021-02-22 10:26:14 +0100  shutoff
minio.service 2021-02-22 18:17:35 +0100  running
```

→ **Instalar minio client** → **mc**.

```
rhel01 ~ # mkdir /opt/minio/mc/bin/ -p
```

```
rhel01 ~ # wget -c https://dl.minio.io/client/mc/release/linux-amd64/mc -O
/opt/minio/mc/bin/mc
```

```
rhel01 ~ # chmod +x /opt/minio/mc/bin/mc
```

```
rhel01 ~ # cp /opt/minio/mc/bin/mc /usr/bin/mc
```

```
rhel01 ~ # mc --autocompletion
```

```
mc: Your shell is set to '/bin/bash', by env var 'SHELL'.
```

```
mc: enabled autocompletion in your 'bash' rc file. Please restart your shell.
```

→ **Instalar Python SDK**.

```
////////////////////////////////////
Instalo el envoltorio → 'virtualenvwrapper' por su sencillez y para realizar 'tests' sencillos y
respecto a los 'buckets' distribuidos que queremos analizar.
```

```
-- Es una mala idea hacerlo como 'root'. Pero esto es un laboratorio ! --
```

```
rhel01 ~ # dnf install @python38
```

```
rhel01 ~ # pip3 completion --bash >> ~/.bashrc
```

```
rhel01 ~ # source .bashrc
```

```
rhel01 ~ # pip3.8 install virtualenvwrapper --user
```

```
rhel01 ~ # which virtualenvwrapper.sh
```



```
/usr/local/bin/virtualenvwrapper.sh
```

```
rhel01 ~ # which python3.8
```

```
/usr/bin/python3.8
```

```
rhel01 ~ # dnf module disable python36
```

```
rhel01 ~ # yum module list python36
```

```
Updating Subscription Management repositories.
```

```
Última comprobación de caducidad de metadatos hecha hace 1:55:28, el mié 24 feb 2021 05:45:42 CET.
```

```
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
```

Name	Stream	Profiles	Summary
python36	3.6 [d][x]	build, common [d]	Python programming language, version 3.6

```
Leyenda: [d] predeterminado, [e] activo, [x] inactivo, [i] instalado
```

```
rhel01 ~ # yum module list python38
```

```
Updating Subscription Management repositories.
```

```
Última comprobación de caducidad de metadatos hecha hace 1:55:41, el mié 24 feb 2021 05:45:42 CET.
```

```
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
```

Name	Stream	Profiles	Summary
python38	3.8 [d][e]	build, common [d] [i]	Python programming language, version 3.8

```
Leyenda: [d] predeterminado, [e] activo, [x] inactivo, [i] instalado
```

```
rhel01 ~ # vim .bashrc
```

```
...
```

```
## virtualenvwrapper.
```

```
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.8
```

```
source /usr/local/bin/virtualenvwrapper.sh
```

```
rhel01 ~ # source .bashrc
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/premkproject
virtualenvwrapper.user_scripts creating /root/.virtualenvs/postmkproject
virtualenvwrapper.user_scripts creating /root/.virtualenvs/initialize
virtualenvwrapper.user_scripts creating /root/.virtualenvs/premkvirtualenv
virtualenvwrapper.user_scripts creating /root/.virtualenvs/postmkvirtualenv
virtualenvwrapper.user_scripts creating /root/.virtualenvs/prermvirtualenv
virtualenvwrapper.user_scripts creating /root/.virtualenvs/postrmvirtualenv
virtualenvwrapper.user_scripts creating /root/.virtualenvs/predeactivate
virtualenvwrapper.user_scripts creating /root/.virtualenvs/postdeactivate
virtualenvwrapper.user_scripts creating /root/.virtualenvs/preactivate
virtualenvwrapper.user_scripts creating /root/.virtualenvs/postactivate
virtualenvwrapper.user_scripts creating /root/.virtualenvs/get_env_details
```

```
rhel01 ~ # env | grep -i VIRTUAL
```

```
VIRTUALENVWRAPPER_WORKON_CD=1
```

```
VIRTUALENVWRAPPER_HOOK_DIR=/root/.virtualenvs
```

```
WORKON_HOME=/root/.virtualenvs
```




```
VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.8
VIRTUALENVWRAPPER_SCRIPT=/usr/local/bin/virtualenvwrapper.sh
VIRTUALENVWRAPPER_PROJECT_FILENAME=.project
```

→ **Realizamos tests** → **'virtualenvwrapper'**.

```
////////////////////////////////////
Creo un entorno virtual con 'virtualenvwrapper', para testear la distribución de los 'buckets', y
por su sencillez Python.
////////////////////////////////////
```

```
rhel01 ~ # mkvirtualenv virtualenv_minio-test -p python3.8
```

```
Using base prefix '/usr'
```

```
New python executable in /root/.virtualenvs/virtualenv_minio-test/bin/python3.6
```

```
Also creating executable in /root/.virtualenvs/virtualenv_minio-test/bin/python
```

```
Installing setuptools, pip, wheel...done.
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/virtualenv_minio-test/bin/predeactivate
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/virtualenv_minio-test/bin/postdeactivate
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/virtualenv_minio-test/bin/preactivate
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/virtualenv_minio-test/bin/postactivate
```

```
virtualenvwrapper.user_scripts creating /root/.virtualenvs/virtualenv_minio-test/bin/get_env_details
```

```
(virtualenv_minio-test) rhel01 ~ # deactivate
```

```
rhel01 ~ # workon virtualenv_minio-test
```

```
(virtualenv_minio-test) rhel01 ~ # pip list
```

```
Package Version
```

```
-----
pip      21.0.1
setuptools 52.0.0
wheel    0.36.2
```

```
(virtualenv_minio-test) rhel01 ~ # pip install minio
```

```
(virtualenv_minio-test) rhel01 ~ # pip install ipython
```

```
(virtualenv_minio-test) rhel01 ~ # pip list
```

```
Package Version
```

```
-----
backcall 0.2.0
certifi 2020.12.5
decorator 4.4.2
ipython 7.20.0
ipython-genutils 0.2.0
jedi 0.18.0
minio 7.0.2
parso 0.8.1
pexpect 4.8.0
pickleshare 0.7.5
pip 21.0.1
prompt-toolkit 3.0.16
```



```
ptyprocess    0.7.0
Pygments      2.8.0
setuptools    52.0.0
traitlets     5.0.5
urllib3       1.26.3
wcwidth       0.2.5
wheel         0.36.2
```

```
(virtualenv_minio-test) rhel01 ~ # ipython
```

```
Python 3.8.3 (default, Aug 18 2020, 08:56:04)
```

```
Type 'copyright', 'credits' or 'license' for more information
```

```
IPython 7.20.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: from minio import Minio
```

```
In [2]: minioClient = Minio('192.168.10.151:9000', access_key='admin',
secret_key='Admin#52', secure=False)
```

```
In [3]: minioClient.make_bucket('bucket-01')
```

```
In [4]: minioClient.make_bucket('bucket-02')
```

```
In [5]: minioClient.make_bucket('bucket-03')
```

```
In [6]: buckets = minioClient.list_buckets()
```

```
In [7]: for bucket in buckets:
```

```
...:     print(bucket.name)
```

```
...:
```

```
bucket-01
```

```
bucket-02
```

```
bucket-03
```

```
In [8]: buckets = minioClient.remove_bucket('bucket-02')
```

```
In [9]: buckets = minioClient.list_buckets()
```

```
In [10]: for bucket in buckets:
```

```
...:     print(bucket.name)
```

```
...:
```

```
bucket-01
```

```
bucket-03
```

→ **Monitorización** → [grafana/prometheus](#)

```
////////////////////////////////////
No lo abordo en esta parte. Pero no es complicado.
////////////////////////////////////
```

<https://grafana.com/docs/grafana/latest/installation/rpm/>

<https://computingforgeeks.com/how-to-install-grafana-on-rhel-8/>

```
rhel01 ~ # vim /etc/yum.repos.d/grafana.repo
```

```
[grafana]
```

```
name=grafana
```

```
baseurl=https://packages.grafana.com/oss/rpm
```

```
repo_gpgcheck=1
```

```
enabled=1
```



```
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

```
rhel01 ~ # dnf update
rhel01 ~ # dnf install grafana
rhel01 ~ # systemctl enable --now grafana-server.service
rhel01 ~ # netstat -nltp
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1027/sshd
tcp6	0	0	:::9000	:::*	LISTEN	2932/minio
tcp6	0	0	:::22	:::*	LISTEN	1027/sshd
tcp6	0	0	:::3000	:::*	LISTEN	3816/grafana-server
tcp6	0	0	:::9090	:::*	LISTEN	1/systemd

```
rhel01 ~ # firewall-cmd --permanent --add-service=grafana
rhel01 ~ # firewall-cmd --reload
```

<http://rhel01:3000>

→ **Instalar 'mc'**

////////////////////
Instalación del cliente 'minio', y su completion.
////////////////////

```
rhel01 ~ # wget -c https://dl.minio.io/client/mc/release/linux-amd64/mc -O /opt/minio/bin/mc
rhel01 ~ # chmod +x /opt/minio/bin/mc
rhel01 ~ # cp /opt/minio/bin/mc /usr/bin/mc
```

```
rhel01 ~ # mc --autocompletion
```

mc: Configuration written to `/root/.mc/config.json`. Please update your access credentials.

mc: Successfully created `/root/.mc/share`.

mc: Initialized share uploads `/root/.mc/share/uploads.json` file.

mc: Initialized share downloads `/root/.mc/share/downloads.json` file.

mc: Your shell is set to '/bin/bash', by env var 'SHELL'.

mc: enabled autocompletion in your 'bash' rc file. Please restart your shell.

```
rhel01 ~ # source .bashrc
```

```
rhel01 ~ # mc mb play/bucket-test
```

Bucket created successfully `play/bucket-test`.

```
rhel01 ~ # mc admin heal play/bucket-test --recursive
```

└─ bucket-test

0/0 objects; 0 B in 2s



Green	2	100.0%	
Yellow	0	0.0%	
Red	0	0.0%	
Grey	0	0.0%	

BIBLIOGRAFIA:

→ **MinIO.**

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-object-storage-server-using-minio-on-ubuntu-18-04-es>
http://tahoot.com/VDO-Data-Reduction-For_linux.pdf
<https://min.io/download#/linux>
<https://medium.com/@diemas.nakayubi/setup-minio-server-on-centos-7-and-proxy-pass-using-nginx-a2dd29428ade>
<https://www.cyberciti.biz/faq/howto-use-wipefs-to-wipe-a-signature-from-disk-on-linux/>
<https://sysadmins.co.za/setup-a-4-node-distributed-minio-cluster/>
<https://docs.debops.org/en/master/ansible/roles/minio/deployment-guide.html>
<https://www.loadbalancer.org/applications/load-balancing-minio-server/>
<https://docs.min.io/docs/python-client-quickstart-guide>
<https://docs.min.io/docs/distributed-minio-quickstart-guide.html>
<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-object-storage-server-using-minio-on-ubuntu-18-04>
<https://min.io/resources/docs/CPG-MinIO-implementation-guide.pdf>
<https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-es>
<https://support.lucidlink.com/support/solutions/articles/31000153348-minio-distributed-4-node-12-disk-erase-coding-bitrot-protected-cluster>
<https://onthedock.github.io/post/170820-probando-minio/>

→ **KVM.**

<https://computingforgeeks.com/how-to-install-kvm-on-rhel-8/>
<https://computingforgeeks.com/how-to-create-and-configure-bridge-networking-for-kvm-in-linux/>
<https://www.garron.me/en/bits/start-stop-kvm-virtual-machine-command-line.html>
https://ravada.readthedocs.io/en/latest/docs/config_console.html
<https://www.certdepot.net/rhel7-access-virtual-machines-console/>
https://www.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.ldva/ldva_r_qemu-imgCommand.html
<https://docs.deistercloud.com/content/Tutorials.100/Linux.80/KVM%20virtualization.40/KVM%20VirSH%20Command.xml?embedded=true>
<https://blog.programster.org/kvm-add-disk>
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_administration_guide/sect-virtualization-virtualized_block_devices-adding_storage_devices_to_guests
<https://blog.programster.org/kvm-add-disk>
<https://www.thegeekstuff.com/2015/02/add-memory-cpu-disk-to-kvm-vm/>



<https://www.unixarena.com/2015/12/linux-kvm-how-to-addressize-virtual-disk-on-fly.html/>
[https://kb.novaordis.com/index.php/Libvirt Storage Volume Renaming](https://kb.novaordis.com/index.php/Libvirt_Storage_Volume_Renaming)
<https://rm-rf.es/convertir-imagenes-vmdk-raw-qcow2-vdi-y-vpc-con-qemu-img/>
<https://linux.dokry.com/kvm-instantneas-externas-en-vivo-y-nombres-de-disco-apropiados.html>

Creative Commons

Reconocimiento-NoComercial-CompartirIgual 3.1 ESPAÑA

© 2021 by carlos briso. Usted es libre de copiar, distribuir y comunicar públicamente la obra y hacer obras derivadas bajo las condiciones siguientes:

a) Debe reconocer y citar al autor original.

b) No puede utilizar esta obra para fines comerciales (incluyendo su publicación, a través de cualquier medio, por entidades con fines de lucro.

c) Si altera o transforma esta obra o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta. Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor. Los derechos derivados de usos legítimos u otras limitaciones no se ven afectados por lo anterior. Licencia completa en castellano.

→ La información contenida en este documento y los derivados de éste se proporcionan tal cual son y los autores no asumirán responsabilidad alguna si el usuario o lector hace mal uso de éstos.