

→ **SOLO Anfitrión:**→ **Comprobamos que la plataforma hardware soporta virtualización:**

```
# grep -e 'vmx' /proc/cpuinfo # Intel systems.
# grep -e 'svm' /proc/cpuinfo # AMD systems.
```

→ **Comprobamos que los módulos KVM han sido cargados:**

```
# lsmod | grep kvm
kvm_intel      315392 0
kvm            847872 1 kvm_intel
irqbypass     16384 1 kvm
```

→ **Instalamos cockpit (no imprescindible):**

```
# dnf install cockpit cockpit-machines
# systemctl start cockpit.socket
# systemctl enable cockpit.socket
# systemctl status cockpit.socket

# firewall-cmd --add-service=cockpit --permanent
# firewall-cmd --reload
```

→ **Acceso al entorno gráfico desde el anfitrión:**

https://SERVER_IP:9090/

→ **Instalamos la virtualización KVM y su visor virt-viewer:**

```
# dnf module install virt
# dnf install virt-install virt-viewer
```

→ **Comprobamos los drivers virt:**

```
# virt-host-validate
QEMU: Verificando for hardware virtualization           : PASA
QEMU: Verificando if device /dev/kvm exists              : PASA
QEMU: Verificando if device /dev/kvm is accessible      : PASA
QEMU: Verificando if device /dev/vhost-net exists        : PASA
QEMU: Verificando if device /dev/net/tun exists          : PASA
QEMU: Verificando for cgroup 'cpu' controller support    : PASA
QEMU: Verificando for cgroup 'cpuacct' controller support : PASA
QEMU: Verificando for cgroup 'cpuset' controller support : PASA
QEMU: Verificando for cgroup 'memory' controller support : PASA
QEMU: Verificando for cgroup 'devices' controller support : PASA
QEMU: Verificando for cgroup 'blkio' controller support  : PASA
```

QEMU: Verificando for device assignment IOMMU support : PASA
 QEMU: Verificando if IOMMU is enabled by kernel : ADVERTENCIA
 (IOMMU appears to be disabled in kernel. Add intel_iommu=on to kernel cmdline arguments)
 QEMU: Verificando for secure guest support : ADVERTENCIA (Unknown
 if this platform has Secure Guest support)

→ **Habilitamos libvirt:**

```
# systemctl enable --now libvirtd.service
# systemctl status libvirtd.service
```

```
# tree /var/lib/libvirt/
```

```
/var/lib/libvirt/
├── boot
├── dnsmasq
│   ├── default.addnhosts
│   ├── default.conf
│   ├── default.hostsfile
│   └── virbr0.status
├── filesystems
├── images
├── network
├── qemu
│   ├── channel
│   │   └── target
│   ├── checkpoint
│   ├── dump
│   ├── nvrpm
│   ├── ram
│   │   ├── libvirt
│   │   └── qemu
│   ├── save
│   └── snapshot
└── swtpm
```

```
# virsh net-list --all
```

```
Nombre Estado Inicio automático Persistente
-----
default activo si si
```

```
# ip addr show dev virbr0
```

```
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group default qlen 1000
    link/ether 52:54:00:a6:de:08 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
```

→ **Preparación para poblado de:** [Discos/Isos/Redes](#) → **Ambiente:** → `--os-variant` → [ol8.2](#)

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

→ **Creación del:** [bridge](#) → [br10](#) **en modo NAT.**

```
# 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='192.168.10.155'/>
    </dhcp>
  </ip>
</network>
```

```
# virsh net-define --help
```

NOMBRE
net-define - define an inactive persistent virtual network or modify an existing persistent one from an XML file

SÍNTESIS
net-define <file>

DESCRIPCIÓN
Define or modify a persistent virtual network.

OPCIONES
[--file] <string> Archivo que contiene una descripción de red XML

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

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

```
# virsh net-list --all
```

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

```
# virsh net-start br10
```

La red `br10` se ha iniciado

```
# virsh net-autostart br10
```

La red `br10` ha sido marcada para iniciarse automáticamente

```
# virsh net-list --all
```

```
Nombre          Estado  Inicio automático Persistente
```

```
-----
br10            activo  si          si
default        activo  si          si
```

```
# ping -c1 192.168.10.5
```

```
PING 192.168.10.5 (192.168.10.5) 56(84) bytes of data.
```

```
64 bytes from 192.168.10.5: icmp_seq=1 ttl=64 time=0.077 ms
```

```
--- 192.168.10.5 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

```
rtt min/avg/max/mdev = 0.077/0.077/0.077/0.000 ms
```

```
# ip addr show dev br10
```

```
4: br10: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
```

```
group default qlen 1000
```

```
link/ether 52:54:00:48:bc:3a 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:**

```
# osinfo-query os | grep -i ol8
```

```
ol8.0          | Oracle Linux 8.0          | 8.0 | http://oracle.com/ol/8.0
ol8.1          | Oracle Linux 8.1          | 8.1 | http://oracle.com/ol/8.1
ol8.2          | Oracle Linux 8.2          | 8.2 | http://oracle.com/ol/8.2
```

→ **Descargamos la ISO correspondiente en:** → `/var/lib/libvirt/Isos`

```
# cd /var/lib/libvirt/Isos/
```

```
# wget -c https://yum.oracle.com/ISOS/OracleLinux/OL8/u4/x86\_64/OracleLinux-R8-U4-x86\_64-dvd.iso
```

→ **Iniciamos la instalación ahora y desde consola:**

```
# virt-install \
```

```
--name oracle-8 \
```

```
--ram 2048 \
```

```
--vcpus 2 \
```

```
--disk path=/var/lib/libvirt/Discos/disco1-oracle8-25GB,size=25,bus=virtio,format=qcow2 \
```

```
--disk path=/var/lib/libvirt/Discos/disco2-oracle8-20GB,size=20,bus=virtio,format=qcow2 \
```

```
--disk path=/var/lib/libvirt/Discos/disco3-oracle8-20GB,size=20,bus=virtio,format=qcow2 \
```

```
--disk path=/var/lib/libvirt/Discos/disco4-oracle8-20GB,size=20,bus=virtio,format=qcow2 \
```

```
--os-variant ol8.2 \
```

```
--os-type linux \
--network bridge=br10 \
--graphics vnc \
--console pty,target_type=serial \
--location /var/lib/libvirt/Isos/OracleLinux-R8-U4-x86_64-dvd.iso \
--extra-args 'console=ttyS0,115200n8 serial'
```

WARNING Gráficos solicitados, pero DISPLAY (pantalla) no está configurado. No se está ejecutando virt-viewer.

WARNING No hay consola que lanzar para el invitado (guest), volviendo al valor predeterminado --wait -1

Empezando la instalación...

Obteniendo archivo vmlinuz... | 9.6 MB

00:00:00

Obteniendo archivo initrd.img... | 72 MB

00:00:00

Asignando 'disco1-oracle8-25GB' | 25 GB

00:00:03

Asignando 'disco2-oracle8-20GB' | 20 GB

00:00:02

Asignando 'disco3-oracle8-20GB' | 20 GB

00:00:02

Asignando 'disco4-oracle8-20GB' | 20 GB

00:00:02

Domain installation still in progress.

Waiting for installation to complete.

^CSe ha interrumpido la instalación del dominio.

Instalación abortada a pedido del usuario

« Interrupción controlada <CTRL> + C

Nos vamos, ...

virsh console --domain oracle-8

Volvemos, ...

...

Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to refresh]: c

=====
Installation

- | | |
|---|--|
| 1) [x] Language settings
(Spanish (Spain)) | 2) [x] Time settings
(Europe/Madrid timezone) |
| 3) [!] Installation source
(Processing...) | 4) [!] Software selection
(Processing...) |
| 5) [x] Installation Destination
(Automatic partitioning
selected) | 6) [x] Kdump
(Kdump is enabled) |
| 7) [x] Network configuration
(Wired (enp1s0) connected) | 8) [x] Root password
(Password is set.) |



9) [] User creation
(No user 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 com_redhat_kdump addon
...
Verifying xcb-util-wm.x86_64 (991/994)
Verifying xdg-utils.noarch (992/994)
Verifying xkeyboard-config.noarch (993/994)
Verifying yajl.x86_64 (994/994)
.
Installing boot loader
..
Performing post-installation setup tasks
[terminated]
...

[OK] Stopped Create System Users.
[OK] Stopped Remount Root and Kernel File Systems.
Stopping Device-Mapper Multipath Device Controller...
[OK] Stopped Device-Mapper Multipath Device Controller.
[OK] Reached target Shutdown.
[OK] Reached target Final Step.
Starting Reboot...
dracut Warning: Killing all remaining processes
Rebooting.
[2251.256604] reboot: Restarting system

virsh list --state-running

Id Nombre Estado

virsh start oracle-8

Se ha iniciado el dominio oracle-8

→ **SOLO Máquina Virtual:**

→ **Ya podemos conectarnos a la máquina virtual:**

ssh root@192.168.10.150

The authenticity of host '192.168.10.150 (192.168.10.150)' can't be established.
ECDSA key fingerprint is SHA256:w2frBiZHRrDBZ1IzNDVhsem1td+MU3QxoZ7eUJSE0gM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '192.168.10.150' (ECDSA) to the list of known hosts.

root@192.168.10.150's password:

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

[root@oracle8 ~]# nmcli c s

```
NAME UUID TYPE DEVICE
enp1s0 56c481ad-4610-4cd1-a472-3a64a7e6509e ethernet enp1s0
```

[root@oracle8 ~]# ip address show dev enp1s0

```
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:dc:61:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.150/24 brd 192.168.10.255 scope global noprefixroute enp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fedc:617b/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

→ **Customizamos nuestra máquina y apagamos (tmux) :**

[root@oracle8 ~]# dnf install tmux

[root@oracle8 ~]# vim .tmux.conf

```
### Keyboard bindings
```

```
### Split windows with easier shortcuts: | and -
```

```
unbind %
```

```
bind | split-window -h
```

```
bind - split-window -v
```

[root@oracle8 ~]# shutdown -h 0

→ **SOLO Anfitrión:**

→ **Creamos un snapshot inicial y clonamos nuestra máquina desde el anfitrión:**

virsh snapshot-create-as --domain oracle-8 --name "Inicial-OK" --atomic

Ha sido creada la captura instantánea Inicial-OK del dominio

virt-clone --original oracle-8 --name oracle-8_clon --auto-clone

Asignando 'disco1-oracle8-25GB-clone' | 25 GB

00:03:51

Asignando 'disco2-oracle8-20GB-clone' | 20 GB

00:00:00

Asignando 'disco3-oracle8-20GB-clone' | 20 GB

00:00:00

Asignando 'disco4-oracle8-20GB-clone' | 20 GB

00:00:00

El clon 'oracle-8_clon' ha sido creado exitosamente.

→ **Podemos editar el xml de nuestra máquina:**

```
# virsh edit --domain oracle-8_clon
```

```
...
```

```
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='/var/lib/libvirt/Discos/disco1-oracle8-25GB-clone'/>
    <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-oracle8-20GB-clone'/>
    <target dev='vdb' bus='virtio'/>
    <address type='pci' domain='0x0000' bus='0x05' slot='0x00' function='0x0'/>
  </disk>
```

```
...
```

→ **Comprobamos snaps, etc, ...**

```
# virsh snapshot-list --domain oracle-8
```

```
Nombre      Hora de creación      Estado
```

```
-----
Inicial-OK  2021-10-04 18:39:49 +0200  shutoff
```

```
# virsh snapshot-list --domain oracle-8 --tree
```

```
Inicial-OK
```

```
# tree /var/lib/libvirt/ -L 2
```

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

```
├── boot
├── Discos
│   ├── disco1-oracle8-25GB
│   ├── disco1-oracle8-25GB-clone
│   ├── disco2-oracle8-20GB
│   ├── disco2-oracle8-20GB-clone
│   ├── disco3-oracle8-20GB
│   ├── disco3-oracle8-20GB-clone
│   ├── disco4-oracle8-20GB
│   └── disco4-oracle8-20GB-clone
├── dnsmasq
│   ├── br10.addnhosts
│   └── br10.conf
```



```
├── br10.hostsfile
├── br10.status
├── default.addnhosts
├── default.conf
├── default.hostsfile
├── virbr0.status
├── filesystems
├── images
├── Isos
│   └── OracleLinux-R8-U4-x86_64-dvd.iso
├── network
├── qemu
│   ├── channel
│   ├── checkpoint
│   ├── dump
│   ├── nvram
│   ├── ram
│   ├── save
│   └── snapshot
├── Redes
│   └── br10.xml
└── swtpm
```

17 directories, 18 files

REFERENCIAS:

<https://www.solvetic.com/tutoriales/article/8362-como-instalar-kvm-en-centos-8-rhel-8/>

Creative Commons

Reconocimiento-NoComercial-CompartirIgual 3.1 ESPAÑA

© 2022 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.