



[Docker Docs](#)

[VirtualBox Docs](#)

[PVE Wiki](#)

[PVE Docs](#)

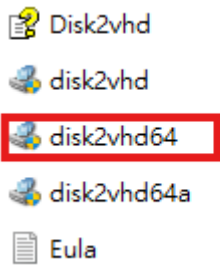
- [PVE Wiki](#)
- [PVE Wiki](#)
- [VirtualBox Wiki](#)



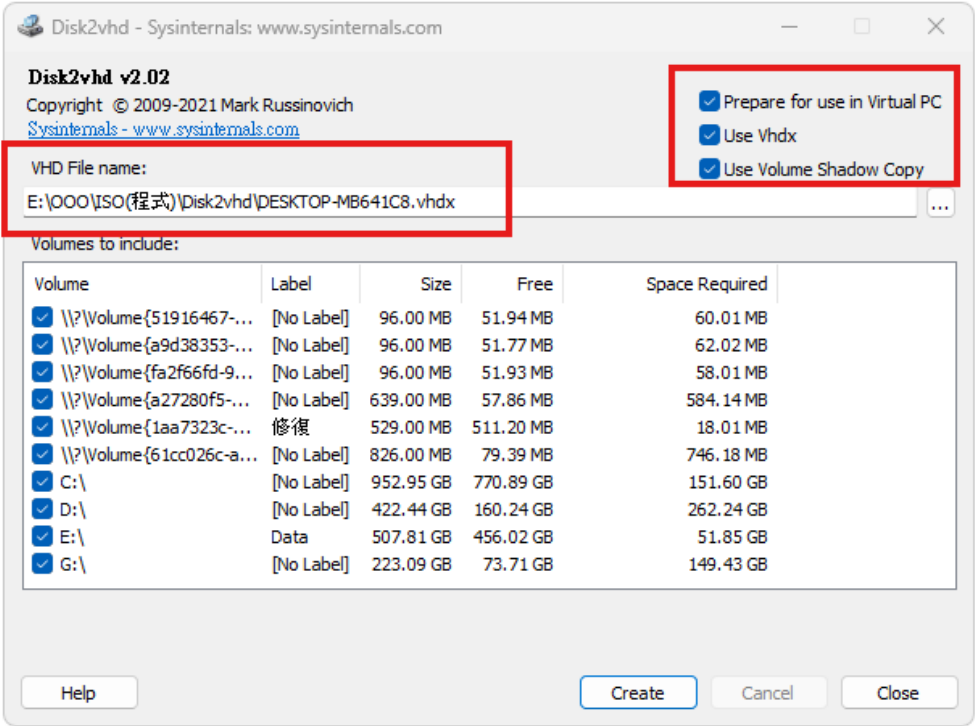
Windowsdisk2vhdvirtualbox

disk2vhd

1. disk2vhd64.exe



3



2. .vhdxVirtualbox(Virtualbox

```
VboxManage.exe clonehd "VHDX.vhdx" "vdi.vdi" -format vdi
```

- 1. Virtualbox
- 2. VDI|VMDK|VHD|RAW

3. Virtualbox



<https://youyouyou.pixnet.net/blog/post/121237292>

PVE ☐ ☐ ☐ ☐

--	--	--	--	--	--	--	--

```
qm unlock <VM ID>
```

```
qm stop <VM ID>
```

backup()

```
vzdump 101 102 107 108 115 114 112 104 110 --compress lzo --mailnotification always --mode snapshot --quiet
1 --storage Backup --mailto mail@domain.com
```

agent

```
apt install qemu-guest-agent
```

node

--	--	--	--	--

```
service rrdcached stop
rm -rf /var/lib/rrdcached/*
service rrdcached start
```

--	--	--	--

service rrdcached status

TASK ERROR: cluster not ready - no quorum?

```

[ ] [ ] [ ] [ ] [ ] pve [ ] [ ] [ ] [ ] [ ] cluster [ ] [ ] [ ] [ ] pve [ ] [ ] [ ] 2 nodes [ ] [ ] [ ] [ ]

```

--	--

```
pvecm status ---[ ]Quorum: 2 Activity blocked
pvecm expected 1 ---[ ][ ][ ][ ][ ][ ]1[ ][ ][ ][ ][ ]/etc/pve
service pve-cluster restart ---[ ][ ][ ][ ][ ]
```

```
TTTTTTT(T)node
```

EX:

--	--	--	--	--	--	--	--

node

```
pvecm delnode (node name)
```

2FA

/etc/pve/user.cfg

raid1 or 5

```
sgdisk -R newdisk olddisk  
sgdisk -G newdisk  
dd if=/dev/olddisk1 of=/dev/newdisk1  
zpool replace rpool olddisk2 newdisk2 (f)
```

pve CVE-2018-3646

```
echo off > /sys/devices/system/cpu/smt/control CPU
```

pve

```
ls -l /dev/disk/by-id/  
vi /etc/pve/qemu-server/xxx(VMID).conf  
virtio0: /dev/disk/by-id/ata-ST2000DL003-9VT166_5YD1ZYK5,size=2000G
```

#

Pve virtio iso

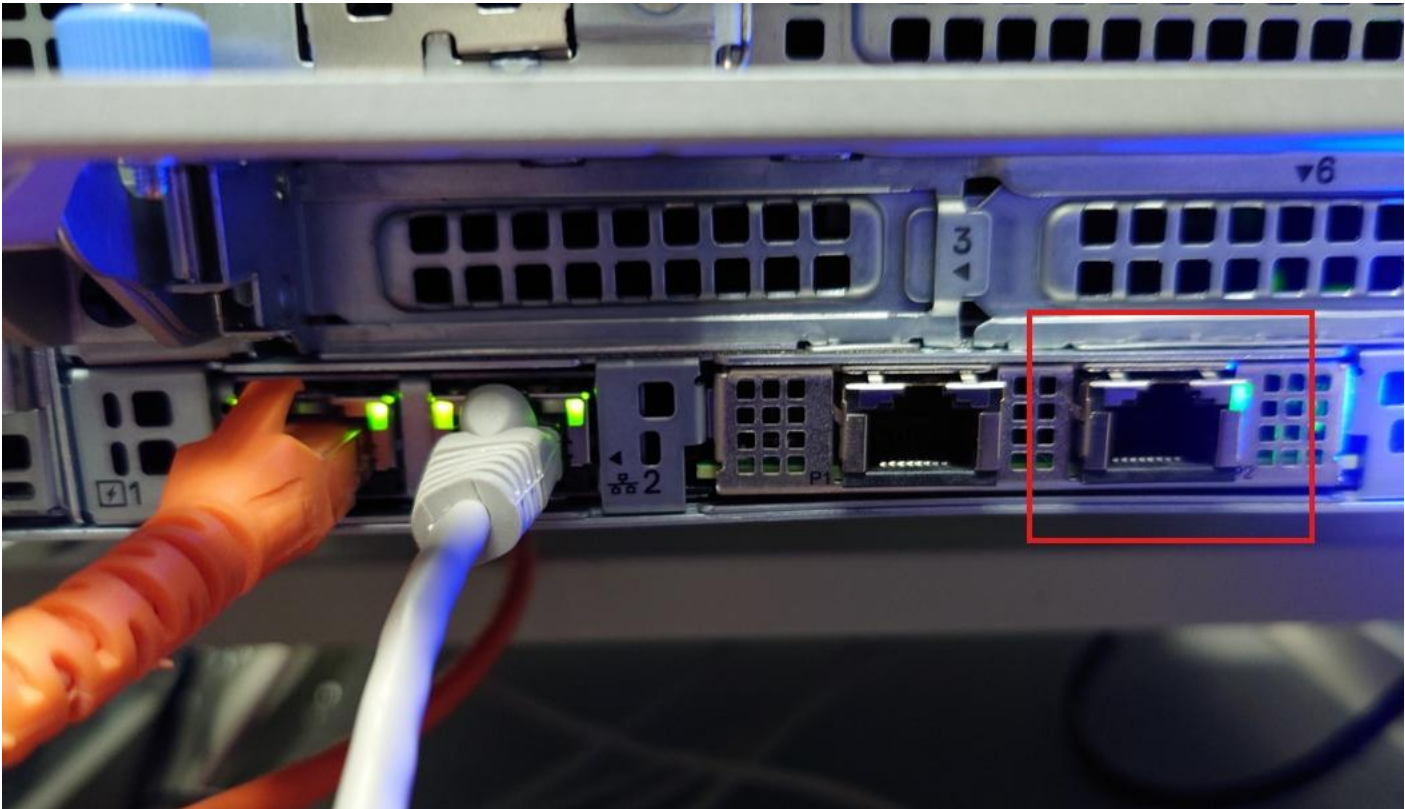
<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/>

ethtool

```
apt install ethtool -y
```



```
ethtool --identify ()
```



00:00:00:00:00:00

00:00:00:00:00:00switch00:00:00

00:00:00 & 00:00
00:00:00:00

```
lspci | grep -i 'eth'
```

00ethtool00:00:00:00:00:00

```
ethtool -i [00:00] #[00:00:00:00:00:00]  
ethtool [00:00] #[00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00]ethtool --identify [00:00] 00:00
```

00:00 & 00:00

00:00:00

BIOS00:00:00:00:00:00VT-d & VMX
00Grub

```
nano /etc/default/grub
```

00:00:00:00:00:00

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"
```

```
pci=
```

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on pci_acs_override=downstream"
```

```
grub
```

```
update-grub
```

```
dd
```

```
img.iso PVE
```

```
nano /etc/pve/qemu-server/[ ].conf
```

```
pve
```

```
qm migrate <ID> <VMID> --online --migration_network x.x.x.0/xx
```

```
PCI
```

```
amd intel cpu grub
```

Intel:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

```
intel_iommu=on
```

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"
GRUB_CMDLINE_LINUX=""
```

AMD:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

```
amd_iommu=on
```

update-grub

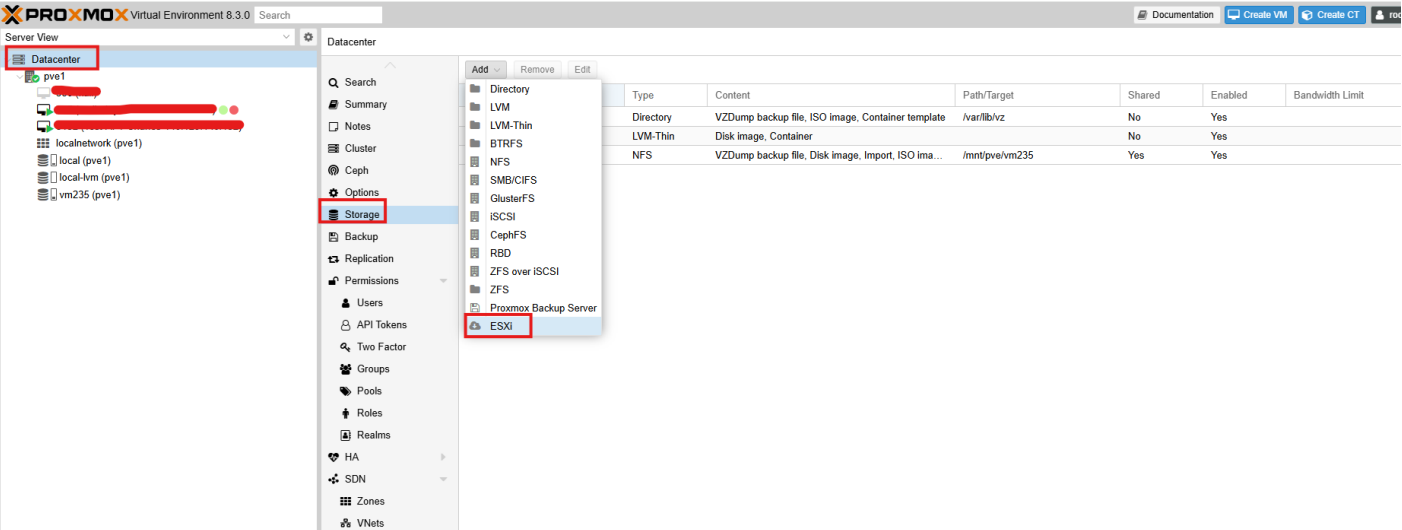
etc/modules

```
#vfio
#vfio_iommu_type1
#vfio_pci
#vfio_virgfd
```

VMWarePVE

1:

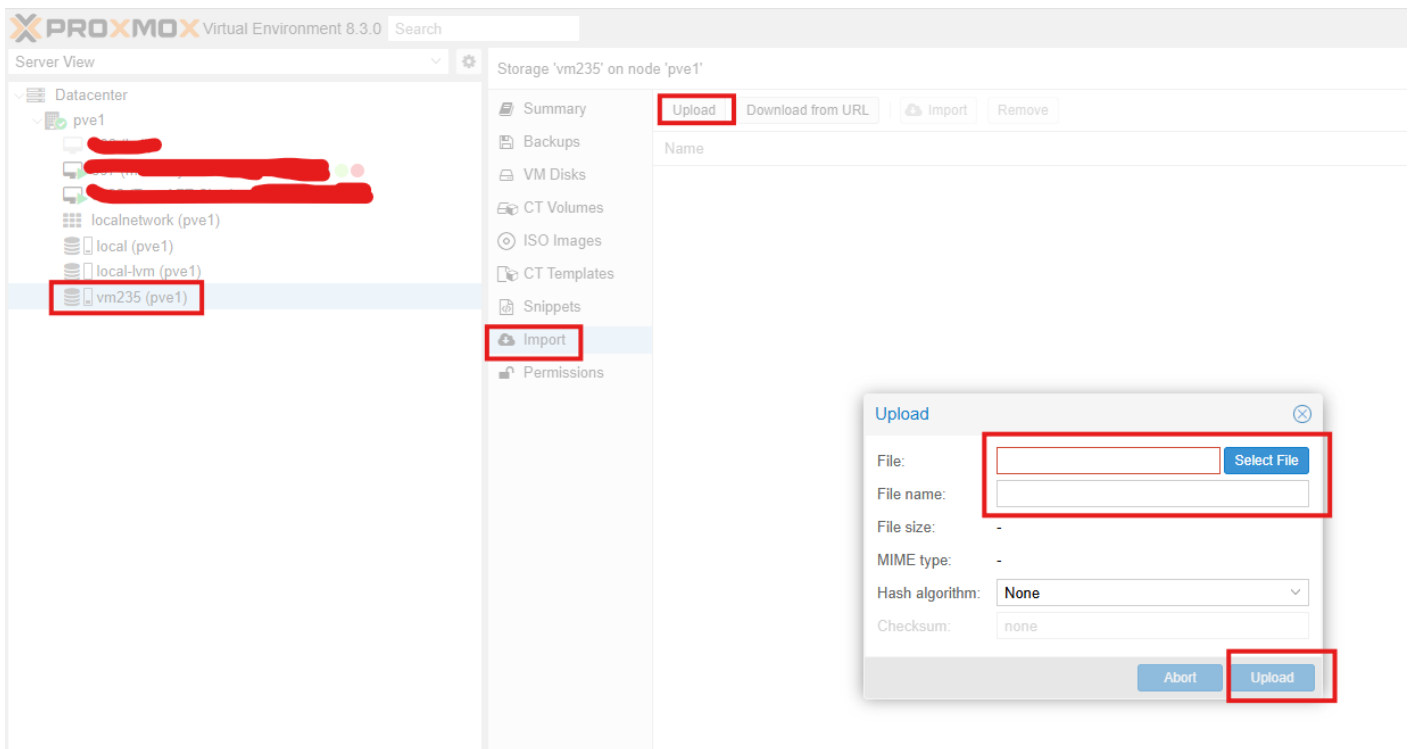
VMWare8.0PVEB.2GUI



VM

2:

GUIOVA PVE



Import PVE

VMWare OVA OVF OVA

```
tar -cvf VM.ova VM.ovf VM.vmdk VM.mf
```

VM

OVF (vmdk mf) PVE

1. scp PVE

```
scp <PVE> root@<IP>:PVE
```

2. qm importovf ovf

```
qm importovf {vmid} {VM name}.ovf {target storage}
```

```
qm --f raw, vmdk, qcow2 --cpu x86-64-v2-AES --scsihw virtio-scsi-
single????
?????GUI???????
```

3. Windows SATA IDE Unset SATA ide

3:

clonezilla qm disk import

```
qm disk import {target VMID} {vmdk file} {target storage}
```

root@qm:~#qm importovf{disk}cpu format{disk}

unset{Windows}sataide

qemu-imgvhdxqcow2

```
qemu-img convert -O qcow2 {disk} {disk}
```

```
qemu-img convert -f {disk} -O {disk} disk disk
```

Local-Lvmlocal

1.local-lvm{disk}

2.cmd{disk}

```
lvremove pve/data
```

```
lvextend -l +100%FREE -f pve/root
```

```
resize2fs /dev/mapper/pve-root
```

3.local{disk}

4.local-lvm{disk}

VirtualBox

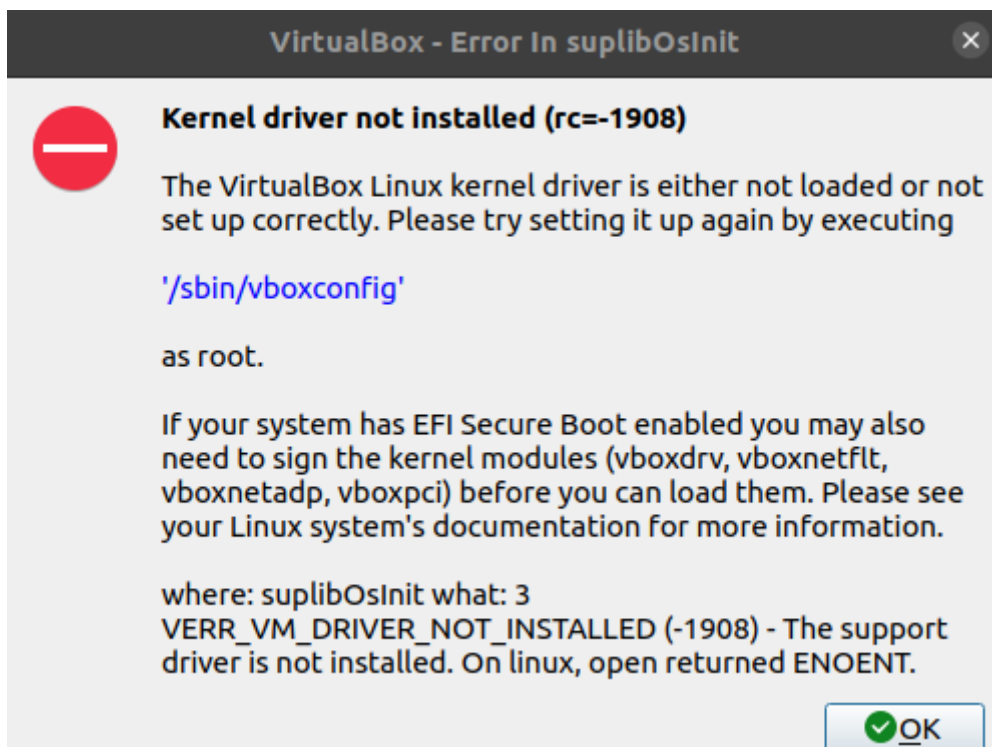
headless

```
VBoxManage startvm "VM|uuid" --type headless
```

```
VBoxHeadless --startvm {VM|uuid}
```



kernel driver not installed(rc=-1908)



root[sudo]@virtualbox

```
sudo /sbin/vboxconfig
```



