

# Proxmox - Migrationen

- Proxmox VMDK oder img in VM importieren
- VM im LVM/ qcow2 format auf anderen Host in ein RBD migrieren ohne zwischenspeicher via SSH
- VM vom RBD auf anderen Host in ein RBD migrieren ohne zwischenspeicher via SSH
- P2V - Laufenden Windows per SSH in Proxmox rbd import
- P2V - Festplatte direkt am Host angeschlossen in Proxmox rbd import
- Cluster Netzwerk wechseln

# Proxmox VMDK oder img in VM importieren

Beschreibung:

Eine neue VM erstellen mit einer VMDK Datei oder img datei.

Vorgang:

Eine neue VM erstellen ohne Disks. Lediglich EFI Disk wenn das System EFI Boot benötigt.  
Die ID der gerade erstellten VM merken

Die VMD oder img Datei auf den Server kopieren z.b per scp.  
Nun mit dem Befehl

```
qm disk import <vmid> <source> <storage> [OPTIONS]
```

Beispiel

```
qm disk import 100 /mnt/meinimage.vmdk data
```

# VM im LVM/ qcow2 format auf anderen Host in ein RBD migrieren ohne Zwischenspeicher via SSH

## Beschreibung

Es gibt Situationen, da möchte man eine VM auf eine andere umziehen lassen. Hat aber kein USB Datenträger bzw. dieser ist zu langsam, bzw man muss vorgänge ausführen, draufkopieren wieder runterkopieren. Auch doppelte Zeit, oder oder oder...

Es gibt die Möglichkeit mit screen und ssh dieses zu bewältigen.

## Vorraussetzungen:

Das die zu kopierende VM natürlich NICHT läuft

Auf dem Zielsystem ist das Ceph soweit vorhanden und auch eingerichtet.

Auf dem Quellsystem ist screen installiert. Befehl dazu

```
apt install screen
```

Quellsystem kann sich per ssh auf dem Zielsystem einloggen.

## Vorgang

### Auf dem Quellsystem Infos holen

Auf dem Quellsystem herausfinden wo unsere LVM Image liegen.

Dazu den Befehl lvs verwenden



## Auf dem Zielsystem Infos holen

Nun auf dem Zielsystem den Pool herausfinden wo die Images hinsollen.  
Dazu auf den Zielsystem einloggen und den Befehl eingeben:

```
ceph osd pool ls
```

Ausgabe, in unserem Beispiel nehmen wir den rbd pool

```
device_health_metrics
rbd
rbd_hdd
cephfs_data
cephfs_metadata
```

Auf dem Zielsystem die nächste freie VMID ermitteln, mittels

```
pvsh get /cluster/nextid
```

Ausgabe:

```
111
```

Somit wird die 111 die neue VM ID auf dem Zielsystem.

## VM Image kopieren

Nun per ssh auf den Quellserver einloggen oder direkt am Server das ist egal.  
nun einen neuen Screen auf machen

Parameter

```
screen -S <namedersession>
```

Beispiel

```
screen -S migrate
```

Nun sind wir in einer Screensitzung, damit wir zum Beispiel unseren Laptop auch ausschalten dürfen.

Denn der Kopierbefehl läuft auf dem Quellsystem jetzt in einer eigenen Session.

## Jetzt den Kopierbefehl ausführen für LVM

#### Parameter

```
ssh root@<zielsystem> "rbd import --image-format 2 - <ceph_rbd_pool>/vm-<NEUE_VMIDAUF_Zielsystem>-disk-<DISKNUMMER>" < /dev/<volumegrup_von_quellsystem>/<localvolume_von_quellsystem>
```

```
ssh root@192.168.178.240 "rbd import --image-format 2 - rbd/vm-111-disk-0" < /dev/datenpool/vm-100-disk-1
```

#### Hinweis:

--image-format 2 ist das RAW Format

## Jetzt den Kopiervorgang ausführen für qcow2

#### Parameter

```
ssh root@<zielsystem> 'rbd import --dest-pool <ceph_rbd_pool> --image-format 2 - vm-<NEUE_VMIDAUF_Zielsystem>-disk-<DISKNUMMER>' < </pfad/zur/image.qcow2>
```

```
ssh root@192.168.178.240 'rbd import --dest-pool rbd_hdd --image-format 2 - vm-116-disk-2' < /mnt/backup/images/105/vm-105-disk-0.qcow2
```

WICHTIG!!!!!! Hierzu auf dem Zielsystem einloggen, denn wir holen uns dann vom quellsystem das image.  
Einfach aus dem screen heraus!

WICHTIG!!!!

#### Parameter

```
ssh root@<quellsystem> 'cat </pfad/zur/image.qcow2>' | rbd import - <ceph_rbd_pool>/vm-<NEUE_VMIDAUF_Zielsystem>-disk-<DISKNUMMER> --image-feature layering
```

#### Beispiel

```
ssh root@192.168.178.2 'cat /mnt/backup/images/105/vm-105-disk-0.qcow2' | rbd import - rbd_hdd/vm-116-disk-2 --image-feature layering
```

Nun läuft der Kopiervorgang und wir können uns nun mit STRG+A gedrückt halten und dann +D von der Sitzung trennen

#### Ausgabe:

```
detached from 7763.migrate]
root@vmserv:~#
```

Um uns wieder mit der Sitzung zu verbinden. Wenn wir noch keine neue Promt sehen für den nächsten Befehl, kopiert er noch.

```
screen -r migrate
```

Augabe:

```
root@192.168.178.240's password:
```

Nun wieder trennen. (STRG+A gedrückt halten dann +D) zum trennen.  
Ab un zu mal schauen wie weit er ist.

Ist der Vorgang abgeschlossen, gibt es diese Fertig Meldung.

```
Importing image: 100% complete...done.
```

Sollte die VM mehrere Disks haben. Den vorgang wiederholen für die nächste DISK.  
Im unseren Beispiel hat die VM aber nur eine DISK.

## Die VM Config Datei übertragen

per scp kann die VM Config übertragen werden.  
Die Dateien werden so benannt<VMID>.conf  
Egal ob es KVM VMs sind oder LXC Conateiner

Die conf Dateien für KVM VMs in das Verzeichnis

```
/etc/pve/qemu-server/
```

und für LXC Conateiner in

```
/etc/pve/lxc/
```

Hat man ein Cluster kopiert man diese Datei auf den jeweiligen node, wo die VM dann hin soll.  
Auf dem Quellsystem einloggen.

Nun der Befehl zum kopieren. In unserem Beispiel ist die vm 100 eine KVM VM

Parameter

```
scp /etc/pve/qemu-server/<altevmid>.conf root@<zielsystem:/etc/pve/qemu-server/<neue_vmid>.conf
```

Beispiel:

```
scp /etc/pve/qemu-server/100.conf root@<zielsystem:/etc/pve/qemu-server/111.conf
```

## VM Conf bearbeiten

Nun ausloggen und auf dem Zielsystem einloggen und die Conf Datei bearbeiten. In unserem Beispiel die VM 111

```
nano /etc/pve/qemu-server/111.conf
```

Ausgabe:

### 111.conf

```
#IP%3A192.168.2.112
boot: cdn
bootdisk: virtio0
cores: 16
memory: 32768
name: Terminal
net0: e1000=12:4E:C7:A9:77:DD,bridge=vibr0
numa: 0
onboot: 0
ostype: win7
sata2: local:iso/Windows10_x64_1909.iso,media=cdrom,size=4091M
sata3: local:iso/virtio-win-0.1.149.iso,media=cdrom,size=316634K
scsihw: virtio-scsi-pci
sockets: 1
virtio0: vms:vm-112-disk-1,cache=writeback,size=500G
virtio1: vms:vm-112-disk-2,cache=writeback,size=1000G
```

Nun den Abschnitt mit den Festplatten

```
...
virtio0: vms:vm-112-disk-1,cache=writeback,size=500G
...
```

auf unseren neuen Datastore abändern. Wir ändern hier den alten Datastore vms zu unserem neuen Datastore rbd ab

```
Parameter
virtio0: <neuer_datastore>:vm-<neuevmid>-disk-<disknummer>,cache=writeback,size=500G

...
virtio0: rbd:vm-111-disk-0,cache=writeback,size=500G
...
```

Sollten mehrere Disks vorhanden sein, dann dieses natürlich für alle Disk wiederholen. Nicht vergessen auch die Disk Nummer am ende ggf anzupassen

Den Abschnitt mit den ISOs äbaändern so das keine mher drin sind.

Vorher

```
...  
sata2: local:iso/Windows10_x64_1909.iso,media=cdrom,size=4091M  
...
```

Nachher

```
...  
sata2: none,media=cdrom  
...
```

Falls die Netzwerkbrücke noch geändert werden muss einfach die vmbr Nummer eingeben. Muss noch ein VLAN hinzugefügt werden. dann sieht das so aus, ansonsten nur die vmbr-ID ändern

Von

```
...  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr0  
...
```

Nach

```
Parameter  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr<ID>,tag=<VLAN_TAG>  
  
...  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr0,tag=102  
...
```

## Fertig

VM Starten glücklich sein.

```
qm start 111
```



# VM vom RBD auf anderen Host in ein RBD migrieren ohne zwischenspeicher via SSH

## Beschreibung

Es gibt Situationen, da möchte man eine VM auf eine andere umziehen lassen. Hat aber kein USB Datenträger bzw. dieser ist zu langsam, bzw man muss vorgänge ausführen, draufkopieren wieder runterkopieren. Auch doppelte Zeit, oder oder oder...

Es gibt die Möglichkeit mit screen und ssh dieses zu bewältigen.

## Vorraussetzungen:

Das die zu kopierende VM natürlich NICHT läuft

Auf dem Zielsystem ist das Ceph soweit vorhanden und auch eingerichtet.

Auf dem Quellsystem ist screen installiert. Befehl dazu

```
apt install screen
```

Quellsystem kann sich per ssh auf dem Zielsystem einloggen.

## Vorgang

### Auf dem Quellsystem Infos holen

Auf den Quellsystem die pool(s) auflisten lassen

```
ceph osd pool ls
```

Ausgabe, in unserem Beispiel nehmen wir den rbd pool

```
.mgr
rbd
rbd_hdd
cephfs_data
cephfs_metadata
```

Auf dem Quellsystem herausfinden wo VM Image liegen im rbd.  
Dazu den Befehl.

```
rbd ls -p <pool-name>

rbd ls -p rbd
```

verwenden

Ausgabe:

```
rbd ls -p rbd
vm-100-disk-0
vm-107-disk-0
vm-109-disk-0
vm-110-disk-0
vm-115-disk-0
vm-117-disk-0
vm-118-disk-0
vm-147-disk-0
```

Uns interessiert die vm-100-disk-0 .

Diese Disk nehmen wir auch für unserer Beispiel.

Wenn man möchte kann man sich noch weitere Informationen vom Image holen.  
ist hier aber optional

```
rbd info -p <pool-name> <image-name>
```

Beispiel:

```
rbd info -p rbd vm-100-disk-0
```

Ausgabe:

```
rbd image 'vm-100-disk-0':
  size 8 GiB in 2048 objects
  order 22 (4 MiB objects)
  snapshot_count: 0
  id: 2e3255c546ee1
  block_name_prefix: rbd_data.2e3255c546ee1
  format: 2
  features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
  op_features:
  flags:
  create_timestamp: Thu Sep  2 14:46:13 2021
  access_timestamp: Thu Sep  2 14:46:13 2021
  modify_timestamp: Thu Sep  2 14:46:13 2021
```

Nun wissen wir das das Ding 8 GB groß ist und können so ne Zeitschätzung fürs kopieren machen.

## Auf dem Zielsystem Infos holen

Nun auf dem Zielsystem den Pool herausfinden wo die Images hinsollen.

Dazu auf den Zielsystem einloggen und den Befehl eingeben:

```
ceph osd pool ls
```

Ausgabe, in unserem Beispiel nehmen wir den rbd pool

```
.mgr
rbd
rbd_hdd
cephfs_data
cephfs_metadata
```

Auf dem Zielsystem die nächste freie VMID ermitteln, mittels

```
pvesh get /cluster/nextid
```

Ausgabe:

```
111
```

Somit wird die 111 die neue VM ID auf dem Zielsystem.

## VM Image kopieren

Nun per ssh auf den Quellserver einloggen oder direkt am Server das ist egal.  
nun einen neuen Screenauf machen

Parameter

```
screen -S <namedersession>
```

Beispiel

```
screen -S migrate
```

Nun sind wir in einer Screensitzung, damit wir zum Beispiel unseren Laptop auch ausschalten dürfen.

Denn der Kopierbefehl läuft auf dem Quellsystem jetzt in einer eigenen Session.

## Jetzt den Kopierbefehl ausführen

Parameter

```
rbid export rbd_hdd/vm-<QUELL_VMID>-disk-<DISKNUMMER> - | pv | ssh -C root@<zielsystem> "rbid import --image-format 2 - <ceph_rbd_pool>/vm-NEUE_VMIDAUF_Zielsystem-disk-<DISKNUMMER>"
```

Beispiel:

```
rbid export rbd_hdd/vm-100-disk-0 - | pv | ssh -C root@192.168.178.5 "rbid import --image-format 2 - rbd/vm-111-disk-0"
```

Hinweis:

--image-format 2 ist das RAW Format

pv -zeigt den fortschritt an

-C steht für Compression an.

Nun läuft der kopiervorgang und wir können uns nun mit STRG+A gedrückt halten und dann +D von der Sitzung trennen

Ausgabe:

```
detached from 7763.migrate]
```

```
root@vmserv:~#
```

Um uns wieder mit der Sitzung zu verbinden. Wenn wir noch keine neue Promt sehen für den nächsten Befehl, kopiert er noch.

```
screen -r migrate
```

Augabe:

```
root@192.168.178.240's password:
```

Nun wieder trennen. (STRG+A gedrückt halten dann +D) zum trennen.  
Ab un zu mal schauen wie weit er ist.

Ist der Vorgang abgeschlossen, gibt es diese Fertig Meldung.

```
Importing image: 100% complete...done.
```

Sollte die VM mehrere Disks haben. Den vorgang wiederholen für die nächste DISK.  
Im unseren Beispiel hat die VM aber nur eine DISK.

## Die VM Config Datei übertragen

per scp kann die VM Config übertragen werden.  
Die Dateien werden so benannt<VMID>.conf  
Egal ob es KVM VMs sind oder LXC Conateiner

Die conf Dateien für KVM VMs in das Verzeichnis

```
/etc/pve/qemu-server/
```

und für LXC Conateiner in

```
/etc/pve/lxc/
```

Hat man ein Cluster kopiert man diese Datei auf den jeweiligen node, wo die VM dann hin soll.  
Auf dem Quellsystem einloggen.

Nun der Befehl zum kopieren. In unserem Beispiel ist die vm 100 eine KVM VM

Parameter

```
scp /etc/pve/qemu-server/<altevmid>.conf root@<zielsystem:/etc/pve/qemu-server/<neue_vmId>.conf
```

Beispiel:

```
scp /etc/pve/qemu-server/100.conf root@<zielsystem:/etc/pve/qemu-server/111.conf
```

## VM Conf bearbeiten

Nun ausloggen und auf dem Zielsystem einloggen und die Conf Datei bearbeiten. In unserem Beispiel die VM 111

```
nano /etc/pve/qemu-server/111.conf
```

Ausgabe:

### 111.conf

```
#IP%3A192.168.2.112
boot: cdn
bootdisk: virtio0
cores: 16
memory: 32768
name: Terminal
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr0
numa: 0
onboot: 0
ostype: win7
sata2: local:iso/Windows10_x64_1909.iso,media=cdrom,size=4091M
sata3: local:iso/virtio-win-0.1.149.iso,media=cdrom,size=316634K
scsihw: virtio-scsi-pci
sockets: 1
virtio0: vms:vm-100-disk-0,cache=writeback,size=500G
```

Nun den Abschnitt mit den Festplatten

```
...
virtio0: rbd:vm-100-disk-0,cache=writeback,size=500G
...
```

auf unseren neuen Datastore abändern. Wir ändern hier den alten Datastore vms zu unserem neuen Datastore rbd ab

```
Parameter
virtio0: <neuer_datastore>:vm-<neuevmid>-disk-<disknummer>,cache=writeback,size=500G

...
virtio0: rbd:vm-111-disk-0,cache=writeback,size=500G

...
```

Sollten mehrere Disks vorhanden sein, dann dieses natürlich für alle Disk wiederholen. Nicht vergessen auch die Disk Nummer am ende ggf anzupassen

Den Abschnitt mit den ISOs äbaändern so das keine mher drin sind.

Vorher

```
...  
sata2: local:iso/Windows10_x64_1909.iso,media=cdrom,size=4091M  
...
```

## Nachher

```
...  
sata2: none,media=cdrom  
...
```

Falls die Netzwerkbrücke noch geändert werden muss einfach die vmbr Nummer eingeben. Muss noch ein VLAN hinzugefügt werden. dann sieht das so aus, ansonsten nur die vmbr-ID ändern

## Von

```
...  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr0  
...
```

## Nach

```
Parameter  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr<ID>,tag=<VLAN_TAG>  
  
...  
net0: e1000=12:4E:C7:A9:77:DD,bridge=vmbr0,tag=102  
...
```

## Fertig

VM Starten glücklich sein.

```
qm start 111
```



# P2V - Laufenden Windows per SSH in Proxmox rbd import

## Beschreibung:

Migration, einen P2V Windows Machine.

Das Windows muss laufen und eine Verbindung zum Proxmox Server muss bestehen / bzw erreichbar sein per ssh.

## Vorraussetzung:

Installiere Cygwin. DD ist schon drin enthalten: <https://www.cygwin.com/> für Alle Benutzer installieren.

Einfach einen Mirror auswählen und weiter weiter. Alle Basis Pakete so lassen und weiter klicken. Dort in den Paketen zusätzlich das programm pv für den Fortschritt anzeigen lassen, nachinstallieren bzw mit auswählen.

## Transferrieren

Cygwin als Administrator starten

Unter /dev/disk/by-id werden die Fetsplatten aufgelistet mit den Partitionen

## Mittels DD kann die Festplatte per SSH kopiert werden

```
dd if=/dev/disk/by-id/nvme-Samsung_SSD_980_PRO_with_Heatsink_1TB_0025_38B2_3140_4F13. bs=4M | pv |  
ssh -C root@192.168.178.9 "rbd import --image-format 2 --image-feature layering - data/vm-804-disk-0"
```

Nun wenn das Image kopiert ist eine Neue VM mit der gleichen ID erstellen und eine Festplatte als dummy.

Diese bekommt dann automatisch die disk-1 am ende.

Nun kann man in die Conf datei gehen

```
nano /etc/pve/qemuserver/<vmid>.conf
```

Den Festplatten eintrag verdoppeln.

Aus der zweiten Festplatte scsi0 dann eine 1

Und bei Festplatte scsi0 (also die erste dann aus disk-1 eine disk-0

Schon hat man zwei Laufwerke drin.

Dann in der gui die disk-1 wieder löschen so das disk-0 übrig bleibt.

nun eventuell mit einer Live CD die virtio Treiber impfen zum Beispiel mit Paragon.

# P2V - Festplatte direkt am Host angeschlossen in Proxmox rbd import

## Beschreibung:

Migration, einer P2V Windows Machine.

Installiert vorher noch die virtio Treiber und die Spcie Geust Tools.

Die Festplatte wird aus dem Computer / Server ausgebaut und direkt an den Server gehängt.

Dann wird die Festplatte als RBD Image ins Ceph kopiert .

## Vorraussetzung:

Installiere Paket pv falls nicht vorhanden

```
apt install pv
```

## Transferrieren

eine Neue VM mit der gleichen ID oder neuen ID erstellen und eine Festplatte als dummy.

Diese bekommt dann automatisch die disk-0 am ende.

Diese Festplatte wieder löschen.

Festplatte ermitteln die per USB oder direktreingehängt wurde ermitteln.

mit

```
lsblk
```

Festplatten auflisten lassen

Unter /dev/disk/by-id werden die Festsplatten aufgelistet mit den Partitionen

## Mittels DD kann die Festplatte per SSH kopiert werden

Die VM id anpassen und die disk nummer.

Tipp: in einer Screen Sitzung arbeiten.

```
dd if=/dev/disk/by-id/nvme-Samsung_SSD_980_PRO_with_Heatsink_1TB_0025_38B2_3140_4F13. bs=4M | pv |  
rbd import --image-format 2 --image-feature layering - data/vm-804-disk-0
```

Nun wenn das Image kopiert ist eine Neue VM mit der gleichen ID erstellen und eine Festplatte als dummy.

Sollten es mehrer disks sein, Eintrag kopieren die disk-1 am ende abändern.

So kommt man in die kann man in die Conf datei

```
nano /etc/pve/qemuserver/<vmid>.conf
```

nun eventuell mit einer Live CD die virtio Treiber impfen zum Beispiel mit Paragon.

# Cluster Netzwerk wechseln

## Beschreibung:

Man möchte in einem Proxmox Cluster das corosync Cluster wechseln, weil man zusätzliche Netzwerkkarten verwenden möchte oder ein VXNetLan gebaut hat oder oder oder...

Auf jeden Fall möchte man neue IP-Adressen setzen.

## Durchführung:

```
nano /etc/pve/corosync.conf
```

Dort die IP-Adressen bei allen Nodes austauschen

```
...
node {
  name: pvetest01
  nodeid: 1
  quorum_votes: 1
  ring0_addr: 10.30.0.10
}
node {
  name: pvetest02
  nodeid: 2
  quorum_votes: 1
  ring0_addr: 10.30.0.11
}
...
```

Wenn zwei Netzwerkkarten konfiguriert waren ist das zweite Ring Adress 1.

Also Ring Adress 0 wenn gewünscht und 1 austauschen

```
nodelist {
  node {
    name: vserv0005
    nodeid: 4
    quorum_votes: 1
  }
}
```

```
ring0_addr: 172.30.128.5
ring1_addr: 172.31.128.5
}
node {
  name: vserv0006
  nodeid: 3
  quorum_votes: 1
  ring0_addr: 172.30.128.6
  ring1_addr: 172.31.128.6
}
node {
  name: vserv0007
  nodeid: 1
  quorum_votes: 1
  ring0_addr: 172.30.128.7
  ring1_addr: 172.31.128.7
```

und die Versionsnummer weiter unten um eine erhöhen  
Versionsnummer erhöhen, alt war 4 neu ist dann eine weiter 5

```
totem {
  cluster_name: hackertest
  config_version: 5
  interface {
    linknumber: 0
```

Nun in der Hosts Datei den proxmox DNS Namen auf die neue Clusteradresse anpassen.  
Alt 192.168.178.2 wurde zu 10.30.10.10. Danach die Hosts neustarten wichtig!

```
127.0.0.1 localhost.localdomain localhost
10.30.10.10 pvetest01.hacker.local pvetest01
```

Hier die Neue Ansicht, alle Quorate mit den neuen Adressen. In diese Beispiel gab es nur link 0 und kein Link 1 dazu

## Cluster Information

Create Cluster **Join Information** Join Cluster

Cluster Name: **hackertest**

Config Version: **5**

Number of Nodes: **4**

## Cluster Nodes

Nodename	ID ↑	Votes	Link 0
pvetest01	1	1	10.30.0.10
pvetest02	2	1	10.30.0.11
pvetest03	3	1	10.30.0.12
pvetest04	4	1	10.30.0.13