

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 runterkoiperen. 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

```
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 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

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

Beispiel:

```
rbd export rbd_hdd/vm-100-disk-0 - | pv | ssh -C root@192.168.178.5 "rbd 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=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-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
```

Version #2

Erstellt: 2 August 2023 10:03:55 von Admin

Zuletzt aktualisiert: 2 August 2023 11:06:12 von Admin