

# Préface

Une VM est une entité un peu abstraite. Dans les faits, cette notion se résume à :

Une forme : un manifest, décrivant le type de hardware présenté à la machine virtuelle par l'hyperviseur (CPU, RAM, nombre de disque, interfaces réseaux)

Un état : des données, sous une forme quelconque, utilisé par l'hyperviseur pour stocker les données persistantes de la VM (le contenu des disques de la VM) Une position face au monde extérieur : une capacité à communiquer avec le monde extérieur

D'autres éléments peuvent être à prendre en compte : règles firewall, affinités, hardware supplémentaire (cdrom etc). Nous n'en parlerons pas ici.

Dans le but de limiter le downtime, la procédure va s'organiser sur quatre grandes parties:

Ce qui peut être préparé longtemps avant Ce qui doit être préparé un peu avant l'intervention Ce qui doit être fait pendant la coupure Ce qui doit être fait après l'intervention Procédure

## Avant l'intervention

Cette phase peut être faite n'importe quand, tant que le manifest source (côté vmware) n'est pas modifié.

### Préparation du manifest

Sur proxmox, créer une VM, incluant l'ensemble des paramètres désirés sauf les disques. Pour faciliter, il est possible de cloner un template migration: la VM 306 (base-migration-vmware).

Sur le clone, on modifiera les paramètres nécessaire: nombre de vcpu, quantité de RAM, interfaces réseaux (nombre et type).

Il est également possible de copier et modifier le manifest suivant, dans `kvm:/etc/pve/qemu-server/<vmid>.conf`:

```
agent: 1
balloon: 4096
boot: dcn
bootdisk: scsi0
cores: 2
hotplug: disk,network,usb,cpu
ide2: none,media=cdrom
memory: 4096
name: base-migration-vmware
net0: virtio=1A:97:F1:B9:A2:2C,bridge=vmbr1,firewall=1,tag=128
net1: virtio=2E:62:E6:7E:78:61,bridge=vmbr1,firewall=1,tag=913
numa: 0
onboot: 1
ostype: l26
scsihw: virtio-scsi-pci
serial0: socket
```

```
shares: 0
smbios1: uuid=d5f35da8-233c-4d53-821e-103c8293794d
sockets: 1
```

## Préparation du réseau

Il convient de s'assurer que les vlan déclarés dans le manifest sont propagés de manière équivalente sur les deux hyperviseurs (vmware et proxmox).

## Peu avant l'intervention

Pour permettre aux deux hyperviseurs d'utiliser le même stockage, nous faisons un storage vmotion sur vmware. Créer un datastore de type NFS, exposé sur vmware, et sur proxmox. Le storage vmotion peut prendre un certain temps (en fonction de la taille des données à copier) Ce datastore est utilisé par Proxmox (storage: migration\_vmware). On peut donc retrouver les données de la VM dans `kvm:/mnt/pve/migration_vmware/`, dans le répertoire portant le nom de la VM (par exemple: `kvm:/mnt/pve/migration_vmware/superVM1/`)

Dans ce même répertoire, on va également trouver un dossier "images", utilisé par proxmox pour stocker des VM.

Pour partager les données de la VM entre vmware et proxmox, nous allons donc nous assurer que ces deux chemins pointent sur les mêmes fichiers:

```
/mnt/pve/migration_vmware/<VMNAME (vmware)> (par exemple:
/mnt/pve/migration_vmware/superVM1)
/mnt/pve/migration_vmware/images/<VMID (proxmox)> (par exemple :
/mnt/pve/migration_vmware/images/528)
```

Un simple lien symbolique fait l'affaire.

Pour finir, rajoutons le ou les disques dans le manifest de la vm côté proxmox, en éditant le fichier `/etc/pve/qemu-server/<vmid>.conf`:

```
scsi0: migration_vmware:<vmid>/filename.vmdk,discard=on
```

Par exemple, pour les trois disques de superVM1, sur la nouvelle VM ID 528:

```
scsi0: migration_vmware:528/superVM1.vmdk,discard=on
scsi1: migration_vmware:528/superVM1_1.vmdk,discard=on
scsi2: migration_vmware:528/superVM1_2.vmdk,discard=on
```

## Intervention ! Attention c'est du lourd !

Si le travail préparatoire a bien été fait, alors la procédure est:

Éteindre la VM côté vmware Démarrer la VM côté proxmox Profit

## Peu après l'intervention

Le service client étant rétabli, nous pouvons nous attarder posément sur la suite des opérations

Migrer les disques proxmox du storage nfs vers un storage rbd: via l'interface: `move disk etc`. Ne PAS cocher la case "delete source" dans Proxmox.

Supprimer la VM dans VMware Supprimer les disques unused, dans le manifest de la VM, sur proxmox (/etc/pve/nodes/<..>/qemu-servers/<vmid>.conf)  
S'assurer que les partitions dans la VM sont bien montés avec le flag discard (/etc/fstab) : si ce n'est pas le cas: modifier /etc/fstab, et faire des "mount -o remount,discard /". Dans tout les cas, passer un coup de fstrim sur l'ensemble des partitions: "fstrim -v /; fstrim -v /home; etc"

Il est impératif de respecter l'ordre de ces instructions.

En effet, pour ceux qui n'ont pas fait attention: suite à l'étape 2, proxmox et vmware voit les mêmes fichiers vmdk. Cela signifie que supprimer un disque (sur proxmox ou sur vmware) va supprimer l'unique copie des données de la VM. Il est donc impératif (j'insiste) de faire des move disk sur rbd AVANT de faire le nettoyage vmware.

Appendice: le cas windows

Forcément, c'est plus compliqué avec windows, et il y a sensiblement plus de downtime.

L'ensemble de la procédure ci-dessus reste la même. On rajoutera cependant certaines étapes.

Dans la phase 1, de préparation, nous allons installer les drivers virtio, disponible ici

Dans la phase 2, nous n'allons pas attacher les disques en scsi, mais en ide. Cela va nous permettre de démarrer windows, lentement mais surement.

Une fois démarré, il va falloir reconfigurer le réseau: remettre les IP etc etc, puis éteindre le serveur.

À ce stade, les disques pourront être mis en scsi, et le reste de la procédure suivie.