

Infrastructure - La virtualisation



Difficulté : Intermédiaire

Notions : Hyperviseurs

I. Introduction

La virtualisation est une technologie qui permet d'exécuter plusieurs systèmes d'exploitations et environnements isolés sur une même machine physique. Optimisant ainsi l'utilisation du matériel et améliorant la flexibilité et la 'scalabilité' des infrastructures informatiques.

Les hyperviseurs sont aujourd'hui devenu un élément essentiel des infrastructures informatiques.

En effet, traditionnellement dans l'informatique, un serveur est une machine physique où l'on va installer un système d'exploitation et des rôles ou des fonctionnalités pour fournir un service aux utilisateurs. Mais avec l'arrivée de la virtualisation, il est devenu possible de détacher les serveurs et services de l'aspect matériel.

Cela apporte certains avantages :

- Réduction des coûts

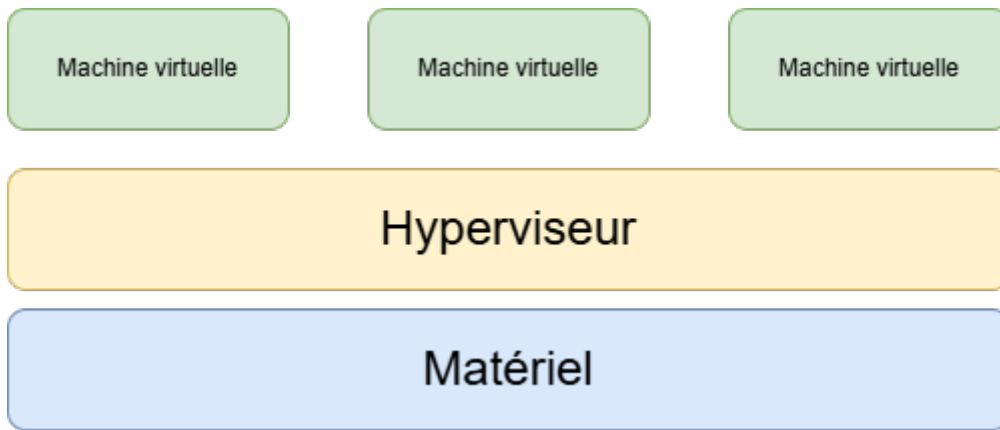
- Réduction des coûts liés au matériel.
- Simplification de la maintenance et des déploiement.
- Flexibilité
 - Les machines peuvent être rapidement déployées ou détruites pour suivre l'évolution du besoin.
 - Elles peuvent être clonées, déplacées sur d'autres matériels physiques ou sites distants.
 - On peut faire évoluer leur matériel afin d'ajuster leurs performances.
 - Création rapide d'environnements de tests ou de formation.
- Consolidation
 - Les machines peuvent être regroupées sur un seul serveur.
 - Avec la haute disponibilité, amélioration de la disponibilité des services.

Info : La virtualisation se démarque de l'émulation dans le fait qu'elle permet de faire fonctionner des machines complètes incluant l'OS et le matériel virtualisé associé, là où l'émulation reproduit logiciellement un matériel différent.

II. Les types d'hyperviseur

2.1 Hyperviseurs de type 1

Les hyperviseur de type 1, aussi appelé **Bare-metal**, sont des hyperviseurs installés directement sur le matériel.



Par exemple : VMware ESXi, Hyper-V, Proxmox VE, Xen...

Cela convient particulièrement à des infrastructures de production.

Pours :

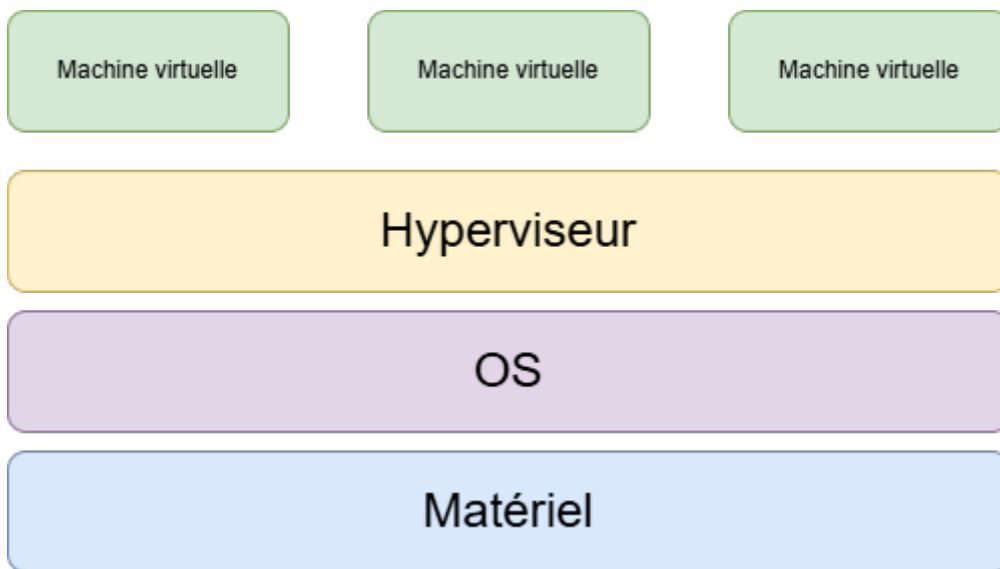
- Performances élevées
- Meilleure isolation
- Fiabilité pour la production
- Possibilité de paramétrer de la haute disponibilité

Contres :

- Plus coûteux (achats de licences).
- Nécessite du matériel spécifique aux performances élevées.
- Nécessite de solides compétences pour une bonne mise en oeuvre.

2.2 Hyperviseurs de type 2

Les hyperviseur de type 2, aussi appelé **Hyperviseur hébergés**, sont des hyperviseurs installés comme un logiciel ou un service par dessus un système hôte existant.



Par exemple : VMware workstation, VirtualBox, Parallels...

Cela convient particulièrement à des environnements de développement, tests ou prototypage.

Pours :

- Simple à installer
- Simple à prendre en main
- Idéal sur poste de travail

Contres :

- Moins d'isolation par rapport au système hôte.
- Moins performant.
- Moins fiables.
- Fonctionnalités réduites.

2.3 Choisir son hyperviseur

En définitive, le choix de l'hyperviseur dépend avant tout de son usage de destination.

Si l'objectif est de monter une infrastructure de production avec une bonne redondance, choisir le type 1.

Sinon, si l'objectif est de disposer d'environnements de tests à des fins de développement ou de prototypage de solutions, privilégier un type 2.

Le choix de la solution dépendra ensuite du budget et des fonctionnalités recherchées.

III. Architecture de plateforme virtualisée

Une architecture de plateforme virtualisée s'articule autour des composantes traitées ci-dessous. Chaque composante aura son rôle et son importance dans certains aspects de la performance ou compatibilité de l'infrastructure.

3.1 Matériel de l'hôte physique

Il s'agit de l'hôte en tant que machine physique sur lequel sera installé l'hyperviseur. Comme toute machine physique, il s'agit d'un serveur matériel avec son hardware.

Le matériel choisi pour le(s) hôte(s) physique(s) influera directement les performances et capacités de l'infrastructure.

Les points à prendre en comptes sont évidemment :

- Le ou les CPU et leur nombre de cœurs
- la RAM
- Le type de stockage (HDD, SSD, hybride, tearing, ...)
- La bande passante des cartes réseaux.

Info : Il est également primordial que la machine choisie dispose des capacité de virtualisation (intel-VT ou AMD-VT).

3.2 L'hyperviseur

L'hyperviseur est le gestionnaire des machines virtuelles. Il permet leur création, manipulation, paramétrage et est également en charge de l'allocation et le suivi des ressources pour les machines virtuelles.

Comme évoqué plus haut, le choix du type de celui-ci dépendra de l'usage.

Le choix du produit quand à lui devra être guidé par les contraintes budgétaires, de performances, mais également les éventuels aspects contractuels.

Comparatif hyperviseurs

Hyperviseur	Type	Fonctionnalités clés	Fourchette budgétaire (indicative)	Mode de licensing
VMware ESXi / vSphere	Type 1	Haute disponibilité (HA), vMotion, DRS, snapshots avancés, gestion centralisée via vCenter, très mature	Moyen à très élevé selon l'édition (Essentials → Enterprise Plus)	Licence par CPU + options (vCenter, fonctionnalités avancées)
Microsoft Hyper-V	Type 1	Intégration Windows Server, Live Migration, Replica, clustering, snapshots, gestion via SCVMM	Faible à moyen (inclus dans Windows Server)	Licence Windows Server (Standard ou Datacenter), par cœur CPU
Proxmox VE	Type 1	KVM + LXC, clustering simple, snapshots, backup intégré, interface web complète, Ceph intégré	Très faible (open-source, support payant optionnel)	Gratuit (AGPL) + abonnement support optionnel
XCP-ng	Type 1	Basé sur Xen, live migration, snapshots, gestion via Xen Orchestra, cluster	Très faible à faible	Gratuit (open-source) + support commercial via Vates
Oracle VM VirtualBox	Type 2	Multi-OS, snapshots, portable, idéal labo/test, large compatibilité	Gratuit	Gratuit (GPL + extension pack sous licence Oracle)
VMware Workstation / Fusion	Type 2	Environnements de test, snapshots, clonage, réseau virtuel avancé	Faible (licence unique)	Licence poste utilisateur

Hyperviseur	Type	Fonctionnalités clés	Fourchette budgétaire (indicative)	Mode de licensing
KVM (Linux)	Type 1 (intégré OS)	Très performant, supporté par la plupart des distributions, snapshots, live migration via libvirt/virt-manager	Gratuit	Inclus dans Linux (GPL), support via distributions

source : copilot, basé sur une liste de sources et de critères définies dans le prompt.

3.3 Les machines virtuelles

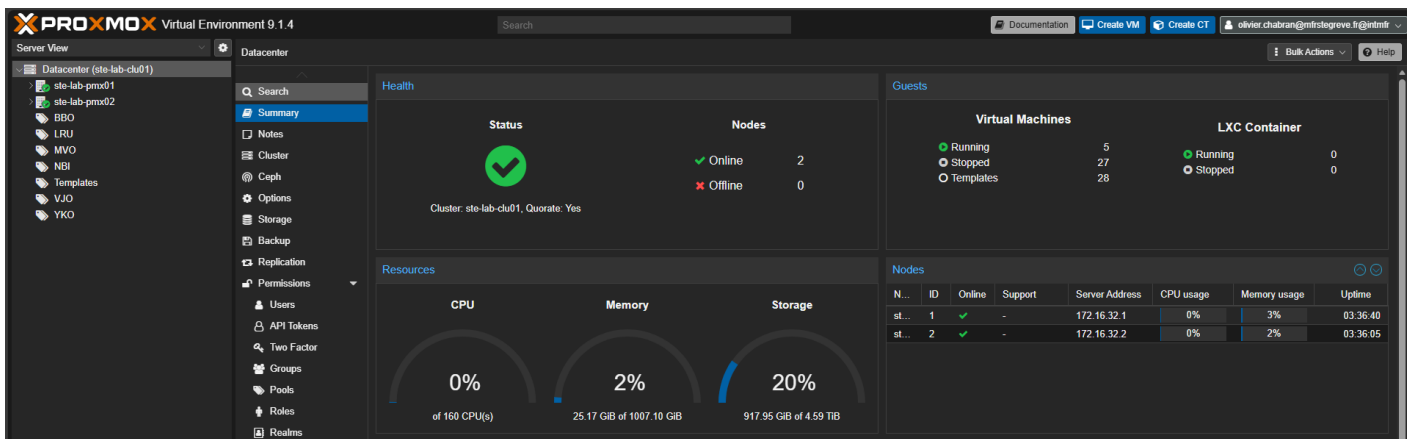
Les machines virtuelles sont les machines hébergées sur l'hyperviseur. Comme toute machine, elles disposent d'un matériel avec ses caractéristiques, sauf que celui-ci n'est pas physique mais plutôt logiciel. Aussi, chaque machine se voit allouée des ressources définies comme suit :

- Vcpu et Vcore (un nombre de socket et de cœurs de CPU que la machine pourra exploiter)
- Mémoire virtuelle (une quantité de RAM maximum allouée à la machine)
- Disques virtuels (souvent sous la forme de fichiers .vmdk, .vhdx, etc.. ; C'est l'espace disque maximal qui sera alloué à la machine)
- Carte réseau virtuelle (une interface réseau pour la connectivité, souvent la gestion des adresses MAC est gérée par l'hyperviseur)

3.4 Les outils de gestion

Afin de gérer le ou les hyperviseur(s) la plupart des solutions intègrent des outils de gestion. Souvent sous la forme d'appliances ou de machines dédiées, ils sont généralement accessibles via une interface web et permettent la configuration du cluster, des hôtes et la gestion des machines virtuelles.

On pourra citer par exemple VMware Vcenter, ProxmoxVE, Hyper-V manager, etc...



3.5 Notions de cluster

Par nature, les machines virtuelles sont donc dé-corrélées du matériel physiques sur lequel elles s'appuient et son donc 'portables'.

On peut ainsi les déplacer d'un hôte à un autre. Cela ouvre la voie à une notion de cluster.

Un cluster est un groupement d'hôtes gérés par un manager qui vont donc pouvoir travailler ensemble afin de se répartir les machines virtuelles. Ce qui permet ainsi :

- La haute disponibilité (les machines peuvent être répliquées et disponibles sur plusieurs hôtes)
- La migration à chaud (les machines peuvent être déplacées d'un hôte à un autre sans forcement les éteindre)
- La répartition de charge (il est possible de répartir les VM afin d'optimiser la consommation de ressources)

IV. Allocation et gestion des ressources

La gestion des ressources est l'un des aspects les plus importants de la virtualisation.

Une VM ne possède pas réellement ses propres composants matériels. Elle reçoit une portion des ressources de l'hôte physique et l'hyperviseur doit en permanence arbitrer entre les besoins de l'ensemble des VM.

Lors de l'allocation de ressources à une VM, il y a plusieurs facteurs à prendre en compte.

4.1 Le CPU

Sur une VM, il est possible d'allouer de la puissance de calcul en ajoutant des Vcpu et des Vcore.

Un vCPU correspond à un thread logique que l'hyperviseur peut planifier. Un vCore est une manière de présenter ces vCPU à la VM.

En définitive, il est possible d'allouer plus de Vcpu ou Vcore à la VM que ce qui est réellement présent sur la machine. Cela est possible car l'hyperviseur sait répartir la charge des instructions sur son matériel physique et que toutes les VM ne sont pas tout le temps à 100% de leur utilisation CPU. Cela s'appelle l'**Overcommit CPU**.

Exemple : un hôte à 64Go de RAM / 16cores, on provisionne 8VM à 4 Vcpu. Il y a overcommit. Cela ne posera pas de problème à moins que la charge sur les machines soit à 100% sur chacune d'elles sur une longue période.

Attention : Contrairement à ce que l'on peut penser, il n'est pas forcément bon d'allouer plus de Vcpu que réellement nécessaire. En effet, cela peut mener à une situation où une VM attend d'exécuter des instructions, mais l'hyperviseur n'a pas de CPU disponibles car déjà alloués aux autres machines. Cela crée donc des tensions sur le CPU et des contentions entre VM. Ce phénomène s'appelle le '**CPU Ready**'.

Un CPU Ready élevé indique :

- surcharge CPU de l'hôte
- trop de vCPU attribués
- VM surdimensionnées
- contention entre VM

Ordres de grandeur :

- < 5 % : normal
- 5-10 % : surveiller
- 10 % : impact visible (latence, lenteur)
- 20 % : problème sérieux

Exemple : Une VM avec 8 cœurs sur un host 4 cœurs --> CPU Ready élevé garanti !

Il est donc important de bien dimensionner les VM pour n'attribuer que le nombre de Vcore nécessaire à son fonctionnement.

4.2 La mémoire

La Mémoire est **LA** ressource critique lorsque l'on parle de virtualisation. En effet, c'est le point sur lequel il est le moins possible de faire de compromis car les services qui sont lancés sur une VM vont consommer une certaine quantité de mémoire au minimum, puis des parts par utilisateurs utilisant le service.

Les hyperviseur vont ensuite utiliser plusieurs mécaniques pour en partager et en optimiser l'usage.

Comme pour le CPU, il est possible d'allouer plus de ressource mémoire que disponible sur l'hôte. Cela s'appelle l'**Overcommit mémoire**.

L'hyperviseur va ensuite se reposer sur plusieurs mécaniques. Dont :

- **Le ballooning**

Lors de l'installation des pilotes de l'hyperviseur sur les systèmes invités, un driver appelé 'balloon' a été installé. Le 'ballon' dans les VM est une portion de la mémoire qui peut être récupérée par l'hyperviseur pour être affectée de manière 'scalable' à d'autres VM.

En cas de tension sur l'hyperviseur, celui-ci 'gonfle' le ballon qui prends la forme d'un processus réservant de la RAM sur la VM invitée. Cette RAM n'est alors plus utilisable par la VM car elle la juge déjà utilisée par ce processus. L'hyperviseur réaffecte donc cette plage de RAM réservée à une autre VM qui en a besoin.

Dès que la tension est libérée, le ballon est 'dégonflé' et la RAM réservée est rendue à la VM.

Info : Bien que le ballooning ne soit pas un problème en soi, il devient problématique dès lors que les VM impactées n'ont plus assez de ressources mémoire pour fonctionner normalement. Cela cause du swapping et donc des lenteurs.

- **La compression**

L'hyperviseur est capable de compresser certaines pages mémoire peu utilisées pour économiser de la mémoire.

- **Swapping hyperviseur**

Tout comme n'importe quelle machine, en cas de sur utilisation de la mémoire, l'hyperviseur est également capable de swap. C'est à dire qu'il va commencer à écrire des pages mémoires sur le disque. Au vu de la dégradation de performances que cela implique, c'est une situation à éviter.

4.3 Le stockage

Le choix du stockage est important et doit être déterminé par les fonctionnalités souhaitées.

Un stockage local en SSD est rapide et offre d'excellentes performances. La sécurité peut en être améliorée avec du RAID.

Attention : Le RAID permet une sécurité locale mais ne permet pas de faire de la **haute disponibilité**.

Pour cela, il faudra opter pour un stockage mutualisé de type NAS ou SAN, ce qui rendra les VM visibles depuis l'ensemble des nœuds du cluster et permettra une bascule rapide en cas de coupure ou maintenance d'un hôte.

Cela permettra aussi d'activer des fonctions d'optimisation de charge et de consommation électrique, ainsi que des règles de répartition des VM.

Exemple : 2 actives directory redondants ne doivent pas se trouver en même temps sur un même nœud.

Si la solution du stockage partagée est privilégiée, il faudra également prendre en compte la liaison réseau avec les hôtes. En effet la bande passante de celle-ci impactera grandement les performances en lecture/Ecriture sur les volumes.

Il est préférable également d'opter pour des cartes réseaux dédiées et optimisées pour cet usage comme les cartes Mellanox



En effet, en plus de leur excellente capacité réseaux, celles-ci intègrent un cache dédié pour le stockage et l'écriture de blocs.

Les transactions sur le stockage se mesure en IOPS (Input/Output per Second). C'est cet indicateur qu'il faut vérifier pour mesurer la performance du stockage.

V. Les fonctionnalités avancées

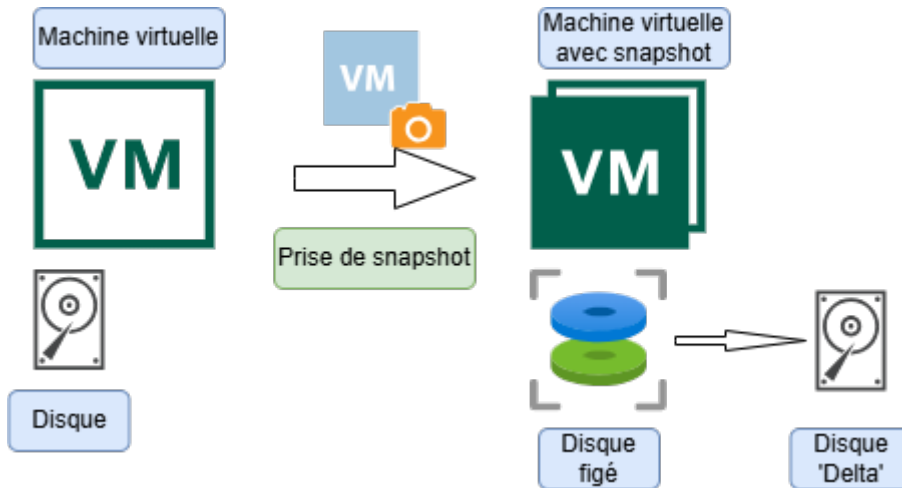
Comme vu plus haut, dans la logique de la virtualisation, les VM ne sont plus directement corrélées au matériel. Cette couche d'abstraction implique que la VM soit vue par l'hyperviseur comme simplement un processus et un ensemble de fichiers.

Cela ouvre la porte à beaucoup de fonctionnalités intéressantes et qui sont devenue clés de la gestion des machines virtuelles au sein d'une architecture d'entreprise.

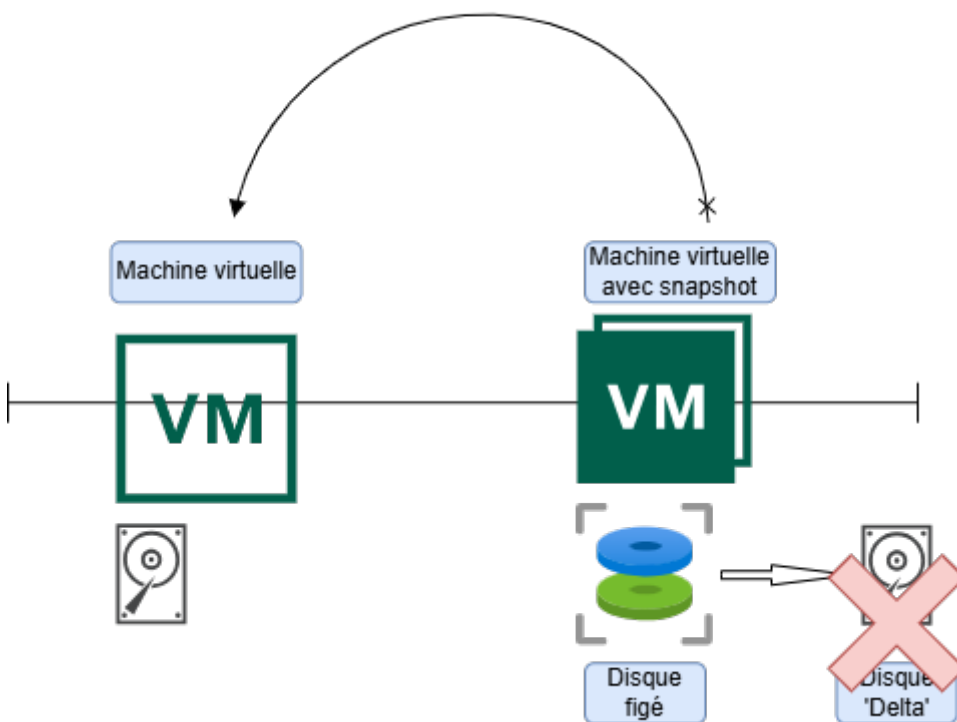
5.1 Les snapshots

Un **snapshot** ou 'cliché instantané', est un point de restauration d'une machine à un instant T.

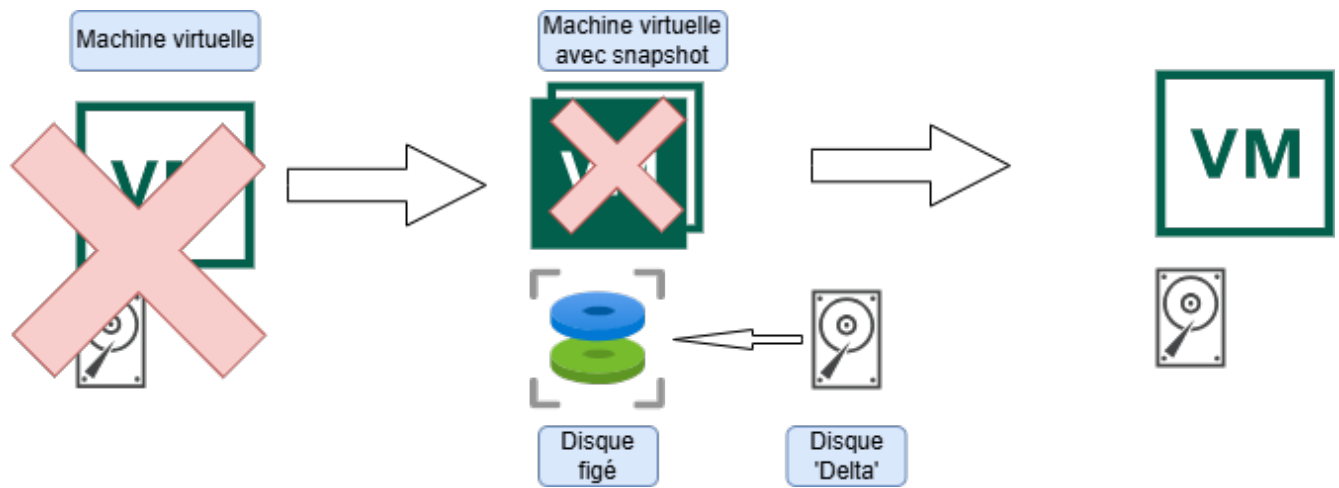
Lors d'une prise de snapshot, le fichier de disque actuel de la VM est figé, un nouveau fichier de disque est créé. Celui-ci contient le Delta depuis le dernier snapshot.



Il sera ensuite possible de faire des modifications, puis de revenir en quelques minutes à cet état précédent. Annulant ainsi tout ce qui est survenu depuis.

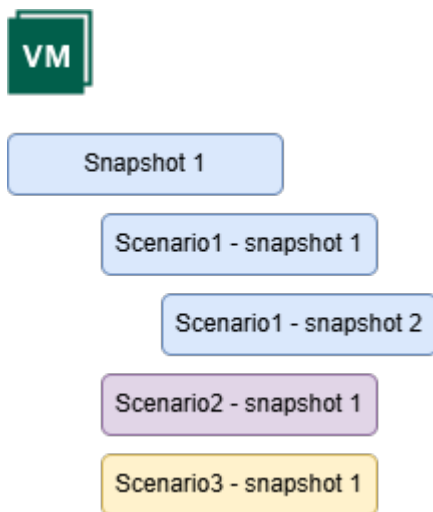


Ou de valider cet état comme définitif et de supprimer les snapshots. Auquel cas, les fichiers du snapshot sont supprimés et les fichiers de disques sont fusionnés.



C'est particulièrement utile pour sécuriser ponctuellement une opération comme une mise à jour, un changement de version ou un test quelconque.

Il est également possible de faire un snapshot, tester un scénario, refaire un snapshot, revenir sur l'original, tester un autre scénario, refaire un snapshot, etc... les snapshots seront ainsi gérés en arborescence.

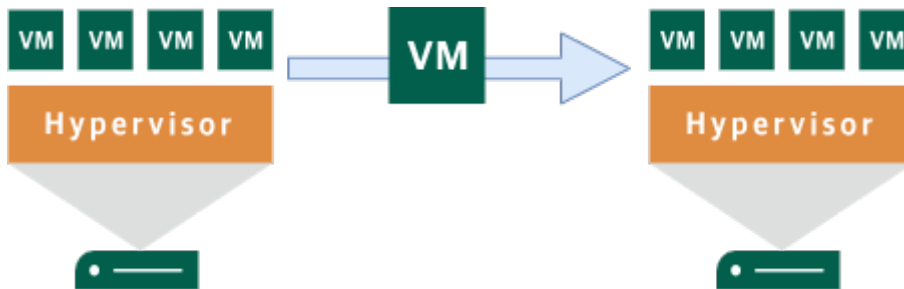


Lors de l'effacement des snapshots, ceux utilisés seront fusionnés, les autres seront détruits. Cette opération s'appelle la '**consolidation**'.

Attention : Les snapshots ne sont pas des sauvegardes. Si cela est utile pour un usage ponctuel, il n'est pas recommandé de les empiler et encore moins de les conserver de façon durable.

5.2 Live migration

Une '**Live migration**' ou migration à chaud est le fait de déplacer une VM d'un hyperviseur ou d'un nœud d'une ferme à un autre, sans forcément avoir besoin d'éteindre cette machine et donc sans interrompre le service.



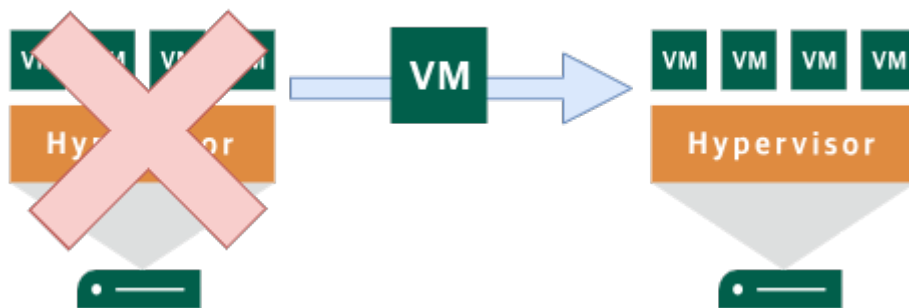
Selon le type de migration, l'hyperviseur fait tout d'abord une copie du fichier descriptif de la VM, puis copie les disques et enfin le fichier mémoire. Dès lors que l'état de la VM est synchronisée sur le nœud source et le nœud de destination, la VM source est coupée et ses fichiers sont détruits.

Dans le cadre de l'utilisation d'un stockage partagé, c'est plus simple dans la mesure où il suffit juste de transférer les fichiers mémoires.

Le déplacement peut être un déplacement sur une autre puissance de calcul, sur un autre stockage voire les deux.

5.3 High availability & Fault Tolerance

La **haute disponibilité** et la **Tolérance à la panne**, sont des mécanismes qui exploitent la capacité de transfert d'une machine d'un hôte à un autre pour permettre automatiquement, en cas de coupure d'un nœud ou d'une VM, d'en reprendre l'exécution depuis un autre nœud.



5.4 Load balancing, fencing

Partant de ce même principe et afin de fiabiliser celui-ci, il est possible de définir des ensembles de règles afin d'éviter que deux machines portant le même rôle dans un but de haute disponibilité se retrouvent au même moment sur le même hôte physique.

Ou inversement, pour des raisons d'économie d'énergie, de regrouper les machines virtuelles sur un même hôte physique afin d'optimiser le rendement énergétique.

5.5 Templates et clones

La dernière fonctionnalité intéressante est de pouvoir, grâce au snapshots et au fait que les machines soient des ensembles de fichiers, est de pouvoir installer une machine, la préparer et la transformer en '**template**' ou modèle.

Cela permet ensuite de repartir de ce modèle afin de faire des déploiement en masse et de standardiser la production. Ce qui représente également un gain de temps non négligeable, car ce qui a été fait une fois pour le template, n'a plus à être refait par la suite.

A partir de ce template, il sera possible de déployer des clones. Il y a deux façons de créer un **clone** :

- Clone lié (la nouvelle machine se base sur les fichiers de l'ancienne et seul le delta est créé)
- Clone intégral (la machine est entièrement recréée à partir du modèle)

La création d'un clone lié à l'avantage d'être immédiate mais l'inconvénient de dépendre des disques des templates pour fonctionner (un peu à la manière des snapshots). En cas de corruption de celui-ci, l'ensemble des clones liées seront rendus inopérants.

La création d'un clone intégral prends plus de temps et de place sur le disque. C'est en effet une nouvelle machine intégralement créée, mais totalement indépendante.

En résumé, les clones liés sont pratique pour faire des tests rapides, mais pour de la production, il faut privilégier les clones intégraux.

5.6 Thin provisioning

Le **thin provisioning**, ou provisionnement dynamique, permet de n'allouer à une machine virtuelle que l'espace disque qu'elle va occuper. La taille définie sur le disque agit ici plus comme une 'limite totale d'espace disponible' que comme une réservation stricte de l'espace disque.

Cette fonctionnalité est très pratique pour des tests et pour optimiser l'espace disque disponible. Mais cela est fortement déconseillé dans un environnement de production car elle entraîne un risque de sur-provisionnement.

En effet, l'espace disque côté hyperviseur est perçu comme libre. Mais si les machines consomment toutes la capacité de leur disque au maximum, la capacité maximale de l'espace de stockage peut être dépassée, ce qui conduit au remplissage à 100% de celui-ci et potentiellement à une corruption des données.

Exemple : 10 VM de 100Go en thin provisioning sur un datastore de 500Go --> Si les VM se mettent à utiliser tout le stockage, risque de corruption et de perte de données.

VI. Virtualisation du stockage

- Datastores
- SAN / NAS / iSCSI / NFS
- vSAN / Ceph / stockage distribué
- Impact des performances disque sur les VM

VII. Virtualisation réseau

- vSwitch / vNIC
- VLAN, trunking, tagging
- Distributed Virtual Switch
- SDN (Software Defined Networking) : principes de base

VIII. Sécurité dans un environnement virtualisé

- Isolation des VM
- Risques spécifiques (VM escape, snapshots oubliés, sprawl)
- Sécurisation de l'hyperviseur
- Gestion des accès et rôles (RBAC)
- Chiffrement des VM / disques / migrations

IX. Administration et bonnes pratiques

- Planification des ressources
- Monitoring (CPU ready, IOPS, latence, mémoire)
- Sauvegarde des VM (image-level, agent-based)
- Gestion du cycle de vie des VM
- Documentation et gouvernance

X. Virtualisation et Cloud

- Virtualisation vs Cloud
 - IaaS, PaaS, SaaS
 - Conteneurs et orchestrateurs (Docker, Kubernetes)
 - Notion de virtualisation “cloud-native”
-

Revision #2

Created 2026-07-17 20:23:57 UTC by Olivier

Updated 2026-07-17 20:25:05 UTC by Olivier