

## Résumé de cours virtualisation et stockage

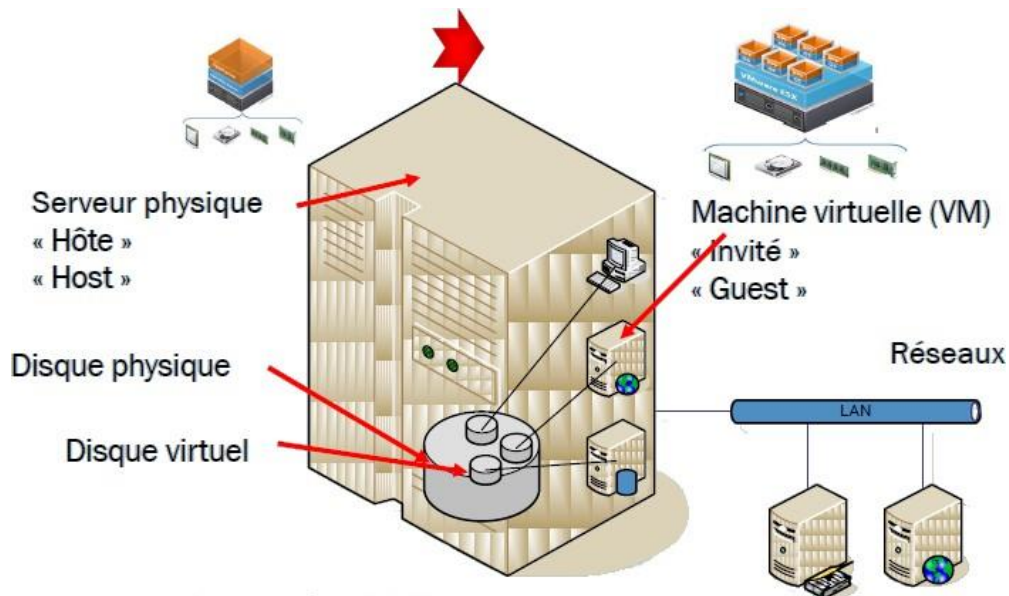
### Table des matières

I.	Définition.....	2
II.	Les domaines de la Virtualisation .....	2
1.	La virtualisation d'application .....	3
2.	La virtualisation de réseaux .....	3
3.	Virtualisation de stockage.....	4
4.	Virtualisation de serveurs .....	5
III.	Les avantages et inconvénients de la virtualisation.....	5
IV.	Les 3 méthodes de virtualisation .....	6
1.	L'isolateur .....	6
2.	L'émulateur .....	6
3.	L'hyperviseur .....	7
V.	Les différents types de virtualisation .....	8
1.	La virtualisation complete .....	8
VI.	La conteneurisation .....	9
1.	Définition .....	9
2.	Les termes cgroups et namespaces .....	10
3.	Différences entre la virtualisation matérielle et la conteneurisation .....	10
4.	LXC et LXD.....	12
5.	Docker .....	12

# I. Définition

La virtualisation désigne un ensemble de techniques permettant de faire tourner plusieurs systèmes d'exploitation (machines virtuelles) sur une même machine physique (hôte).

Le principe de la virtualisation est donc le partage ou la mutualisation des ressources d'un serveur.



## La Virtualisation repose sur trois éléments importants :

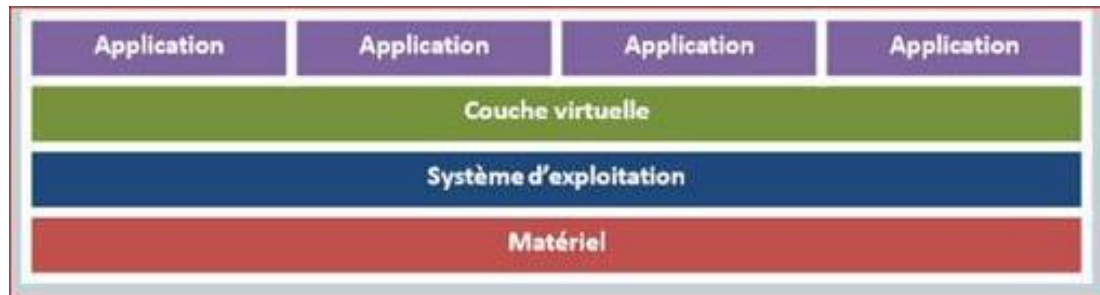
- L'abstraction des ressources informatiques ;
- La répartition des ressources par l'intermédiaire de différents outils, de manière que celles-ci puissent être utilisées par plusieurs environnements virtuels ;
- La création d'environnements virtuels.

## II. Les domaines de la Virtualisation

1. La virtualisation d'application
2. Virtualisation de réseaux
3. Virtualisation de stockage
4. Virtualisation de serveurs

## 1. La virtualisation d'application

La virtualisation d'applications est une technologie logicielle qui va permettre d'améliorer la portabilité et la compatibilité des applications en les isolant du système d'exploitation sur lequel elles sont exécutées. Elle consiste à encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné.



La couche virtuelle ajoute des avantages au système virtualisé en permettant d'exécuter des applications conçues pour d'autres systèmes. A titre d'exemple, le logiciel Wine permet d'exécuter certains programmes Windows sous Ubuntu.

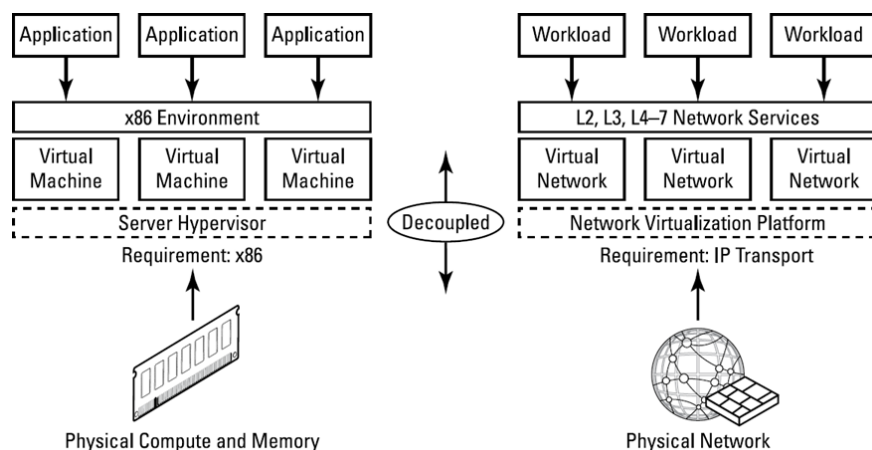
La virtualisation d'applications peut aussi consister à rendre disponible une application installée sur un serveur pour un client.

**Parmi les principaux acteurs de la virtualisation, on peut citer :**

- Citrix XENAPP 6,
- Microsoft APP-V
- VMWare ThinAPP.

## 2. La virtualisation de réseaux

La virtualisation de réseaux permet aux applications de s'exécuter sur un réseau virtuel comme si c'était un réseau physique. Avec un réseau virtuel, les fonctions de commutation, le routage, le contrôle d'accès, le pare-feu, la qualité de service (QoS) et l'équilibrage de charge sont implémentées dans le logiciel. La virtualisation de réseau fait passer l'intelligence du matériel dédié au logiciel flexible.



### 3. Virtualisation de stockage

Dans une machine virtuelle, les données sont stockées sur un disque dur virtuel. Ce disque dur se présente sous forme de fichier dans le système de fichiers de l'hôte :

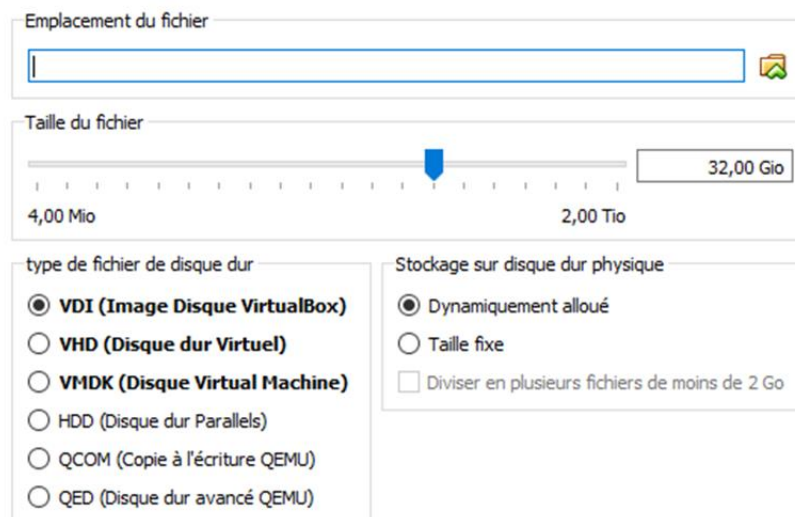
- VHD chez Microsoft
- VDI chez Oracle
- VMDK chez VMWare
- OVF format ouvert

Les disques virtuels peuvent être statiques ou dynamiques.

Dans le cas où **le disque est statique**, si on crée un disque de 50 Go, le fichier de disque virtuel fera 50 Go sur le système hôte.

Avec un **disque dynamique**, le fichier de disque virtuel se remplit au fur et à mesure qu'il est utilisé. Un disque de 50 Go dans lequel il n'y a pas de données ne pèsera dans le système de fichiers hôte pas grande chose (voir figure).

← Créer un disque dur virtuel



Grâce à la virtualisation de stockage, il est possible de réallouer dynamiquement de l'espace de stockage. Ainsi, un serveur nécessitant un espace de stockage supplémentaire pourra rechercher des ressources non allouées sur le disque logique. Inversement, un serveur nécessitant moins d'espace de stockage pourra libérer cet espace et le rendre disponible pour d'autres serveurs.

#### La virtualisation de stockage permet aussi :

- D'ajouter un périphérique de stockage supplémentaire sans interruption des services ;
- De regrouper des unités de disques durs de différentes vitesses, de différentes tailles et de différents constructeurs.

La virtualisation du stockage touche aussi les éléments de stockage dédié, comme les NAS ou SAN.

## 4. Virtualisation de serveurs

La virtualisation de serveur est un principe permettant de faire fonctionner simultanément, sur un seul serveur physique, plusieurs serveurs virtuels. Cette technique permet aux entreprises d'utiliser des serveurs virtuels en lieu et place de serveurs physiques. Si cette virtualisation est faite au sein de la même entreprise, le but est de mieux utiliser la capacité de chaque serveur par une mise en commun de leur capacité.



### **La virtualisation de serveurs permet de :**

- Regrouper plusieurs serveurs physiques sous-employés sur un seul hôte qui exécute des systèmes virtuels ;
- Réduire la surface au sol, les équipements matériels, le besoin de climatisation et le nombre d'administrateurs ;
- Réaliser des économies (locaux, consommation électrique, personnel) ;
- Réduire les délais de mise à disposition de nouveaux serveurs ;
- Simplifier l'administration et la gestion ;
- Améliorer le niveau de service et la disponibilité des applications ;
- Simplifier la migration des applications sur de nouveaux serveurs ;
- Mettre en place un PRA (plan de reprise d'activité).
- S'inscrire dans la démarche Green IT.

## III. Les avantages et inconvénients de la virtualisation

### **Les avantages de la virtualisation sont nombreux :**

- Limiter le gaspillage de ressources,
- Réduire les coûts

### **Les inconvénients :**

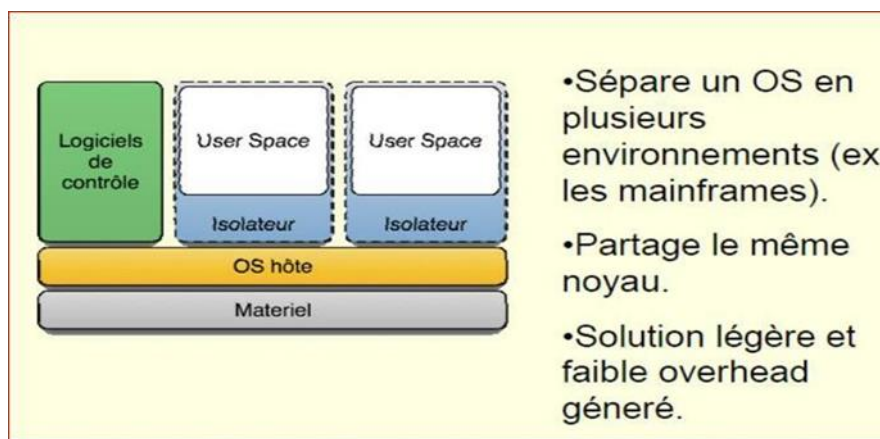
- Bien que la virtualisation soit implémentée sur des machines puissantes, elle peut réduire les performances des applications. Suivant le type de virtualisation envisagé, cette perte de performances peut ou non être significative.

## IV. Les 3 méthodes de virtualisation

1. L'isolateur
2. L'émulateur
3. L'hyperviseur

### 1. L'isolateur

Un isolateur est installé dans un système d'exploitation existant. Il permet de cloisonner des applications qu'on souhaite virtualiser dans des zones d'exécution mémoire différentes. Ces zones génèrent un contexte propre à chaque application. L'isolation n'est pas une technique de virtualisation au sens propre du terme. L'isolation permet d'obtenir des environnements qui semblent se comporter comme des VM, mais qui partagent un même noyau. On parle ici de conteneur. Un conteneur fait tourner une distribution sur une distribution.



### 2. L'émulateur

Dans l'émulation, la machine physique « hôte » héberge de multiples VMs en leur donnant accès à ses ressources matérielles de façon optimale et selon des règles de partitionnement ajustables.



### Avantages :

- Facilité de mise en œuvre et d'utilisation ;
- Très bonne compatibilité d'OS.

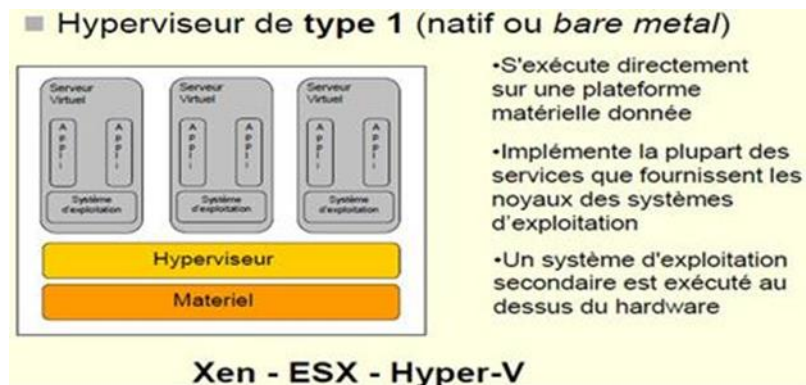
### Inconvénients :

- Mauvaises performances, matériel émulé.

## 3. L'hyperviseur

Un hyperviseur est un système d'exploitation particulier qui va héberger lui-même un ou plusieurs systèmes d'exploitation. L'hyperviseur alloue aux machines virtuelles des ressources matérielles. Il existe des hyperviseurs de type 1 et de type 2.

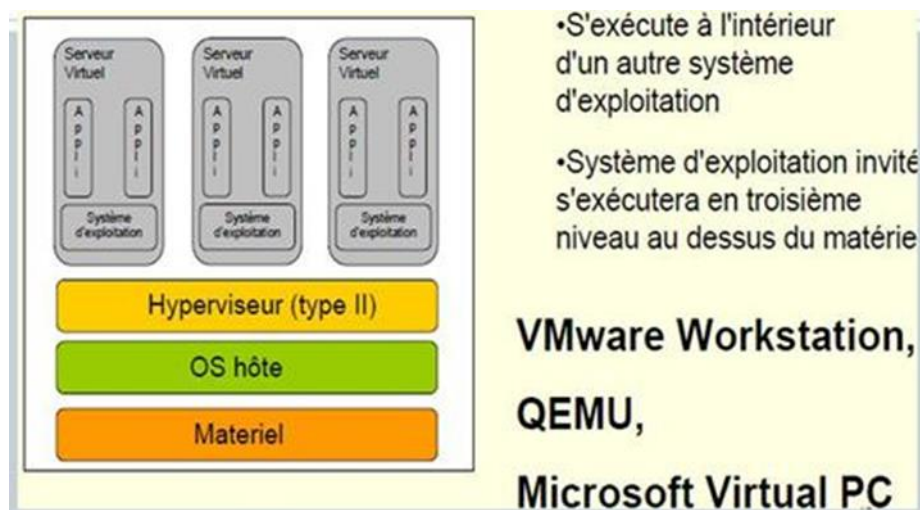
### **L'hyperviseur de type 1 (natif ou bare metal)**



### Exemples :

- VMware ESXi,
- Hyper-V,
- Proxmox VE,
- Open-VZ,
- XEN,
- KVM.

### **L'hyperviseur de type 2 (ou host-based)**





### **Exemples :**

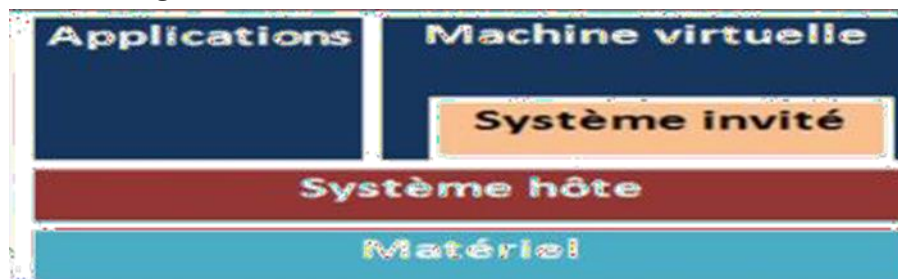
- virtualbox,
- vmware workstation,
- QEMU,
- Microsoft Virtual PC.

## **V. Les différents types de virtualisation**

1. La virtualisation complete
2. La paravirtualization

### **1. La virtualisation complete**

La virtualisation complète (full virtualization), dénommée ainsi par opposition à la paravirtualisation consiste à émuler l'intégralité d'une machine physique pour le système invité. Le système invité « croit » s'exécuter sur une véritable machine physique. Le système invité, ne dialogue jamais directement avec le matériel réel, mais avec l'émulateur. La caractéristique principale de la virtualisation complète est que les systèmes invités n'ont pas à être modifiés pour être utilisés dans une machine virtuelle utilisant une technologie de virtualisation.



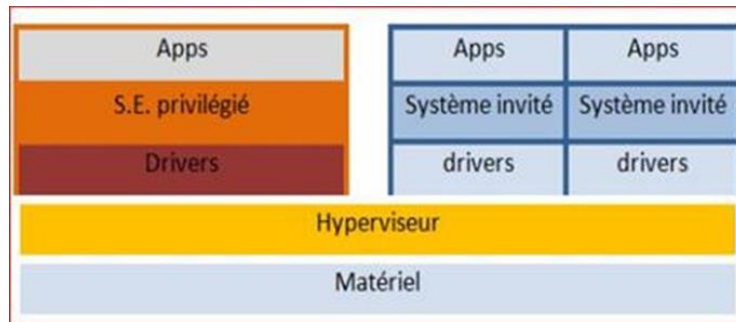
### **Exemples :**

- VirtualBox,
- Microsoft VirtualPC
- Microsoft VirtualServer



## 2. La paravirtualization

La paravirtualisation est très proche du concept de la virtualisation complète, dans le sens où c'est toujours un système d'exploitation complet qui s'exécute sur le matériel émulé par une machine virtuelle. La para-virtualisation évite d'utiliser un système hôte complet pour faire la virtualisation. Les performances sont bien meilleures en paravirtualisation qu'en virtualisation complète.



### Exemples :

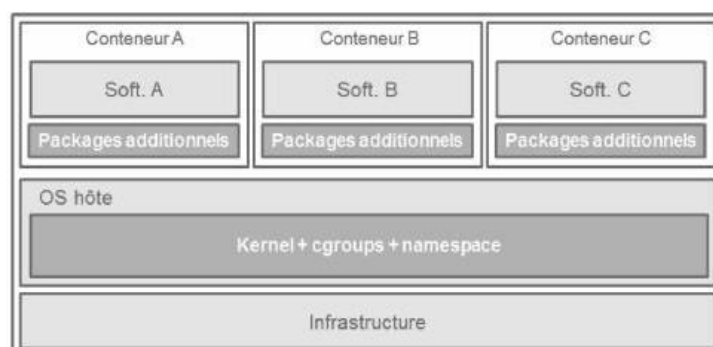
- Xen ,
- KVM ,
- ESX/ESXi ,
- Hyper-V

## VI. La conteneurisation

### 1. Définition

Dans une architecture à base de conteneurs (voir figure), le contenu du conteneur, c'est-à-dire le code et ses dépendances (jusqu'au niveau de l'OS), est de la responsabilité du développeur.

Le conteneur offre l'isolation permettant à un développeur d'embarquer l'ensemble des dépendances logicielles dont il a besoin (y compris les dépendances de niveau OS). De plus, un conteneur s'appuie sur le noyau (kernel) du système d'exploitation hôte. Il est donc très léger et démarre presque aussi vite que le processus qu'il encapsule. Le nombre de conteneurs qu'un même hôte peut exécuter est donc nettement plus élevé que son équivalent en machines virtuelles.



## 2. Les termes cgroups et namespaces

Les termes « **cgroups** » et « **namespaces** » font référence à des extensions du noyau Linux qui rendent possible la réalisation de conteneurs « isolés » les uns des autres.

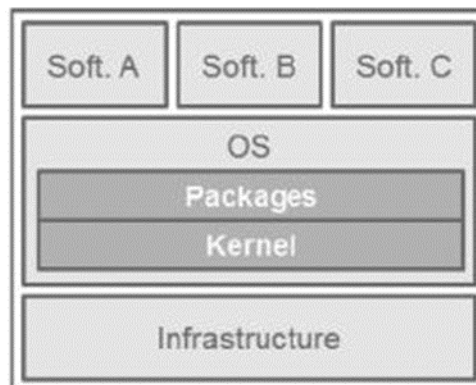
- **CGroups (pour Control Groups)** permet de partitionner les ressources d'un hôte (processeur, mémoire, accès au réseau ou à d'autres terminaux). L'objectif est de contrôler la consommation de ces ressources par processus.
- Les **Namespaces** sont indépendants de CGroups, mais fonctionnent de concert. Ils permettent de faire en sorte que des processus ne voient pas les ressources utilisées par d'autres. Si CGroups gère la distribution des ressources, Namespaces apporte l'isolation nécessaire à la création de conteneurs.

En résumé, un conteneur est tout simplement un système de fichiers sur lequel s'exécutent des processus (de préférence un par conteneur) de manière :

- **contrainte** : grâce à CGroups qui spécifie les limites en termes de ressources ;
- **isolée** : grâce notamment à Namespaces qui fait en sorte que les conteneurs ne se voient pas les uns les autres.

## 3. Différences entre la virtualisation matérielle et la conteneurisation

Étudions l'empilement des couches dans le cas du déploiement de trois logiciels sur un même hôte (host) sans machine virtuelle ou conteneur.



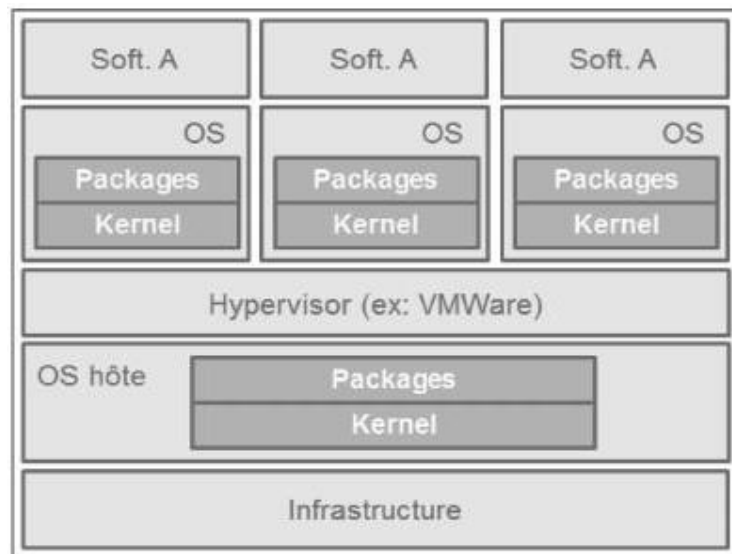
Dans ce type d'installation, on imagine que plusieurs situations problématiques peuvent survenir :

- Les différents logiciels peuvent interagir entre eux s'ils n'ont pas été conçus par le même éditeur. Ils pourraient, par exemple, nécessiter des packages (bibliothèques, extensions) ou des versions de système d'exploitation différentes. Ils peuvent aussi ouvrir des ports réseaux identiques, accéder aux mêmes chemins sur le système de fichiers ou encore entrer en concurrence pour les ressources I/O ou CPU ;

- Toute mise à jour de l'OS hôte va nécessairement impacter tous les logiciels qui tournent dessus ;
- Chaque mise à jour d'un logiciel pourrait entraîner des impacts sur les autres.

L'expérience montre que l'exécution, sur le même système, de logiciels fournis par des éditeurs différents qui n'auraient pas testé cette cohabitation est très souvent problématique.

La virtualisation matérielle offre une réponse appropriée à ces problèmes de cohabitation sans aucun doute. Pour mieux comprendre les limites de celle-ci et cerner l'importance des conteneurs, étudions à présent, le déploiement de trois machines virtuelles sur le même hôte.



Dans les faits, comme nous le voyons sur la figure ci-dessus, cette virtualisation matérielle offre un niveau d'isolation élevé. Chaque logiciel se trouve dans son bac à sable (sandbox en anglais). Les problèmes évoqués précédemment sont donc résolus, mais d'autres apparaissent :

- le poids d'une machine virtuelle est tout d'abord très important. Une machine virtuelle est une machine et, même avec un système d'exploitation minimal, un système d'exploitation moderne consommera difficilement moins de quelques Go de mémoire. La distribution de ce type de package demandera une bande passante réseau conséquente ;
- la machine virtuelle embarque trop d'éléments. Elle ne laisse pas le choix à l'exploitant (selon la manière dont elle aura été configurée) de choisir librement ses caractéristiques, comme le type de stockage, le nombre de CPU utilisés, la configuration réseau. Évidemment, les solutions de gestion d'environnements virtualisés (par exemple, vCenter de VMWare) offrent des solutions, mais celles-ci ont presque toujours des impacts sur le logiciel. Ce dernier ne peut pas être conçu sans savoir comment il va être exécuté.

L'architecture qui offre le bon compromis est celle à base de conteneurs.

Une architecture à base de conteneurs offre une solution de compromis. Comme cela a été dit plus haut, le conteneur offre **l'isolation** permettant à un développeur d'embarquer l'ensemble des dépendances logicielles dont il a besoin (y compris les dépendances de niveau OS).

De plus, un conteneur s'appuie sur le noyau (kernel) du système d'exploitation hôte. Il est donc très léger et démarre presque aussi vite que le processus qu'il encapsule. Le nombre de conteneurs qu'un même hôte peut exécuter est donc nettement plus élevé que son équivalent en machines virtuelles.

## 4. LXC et LXD

**LXC, contraction de l'anglais Linux Containers** est un système de virtualisation, utilisant l'isolation comme méthode de cloisonnement au niveau du système d'exploitation. Il est utilisé pour faire fonctionner des environnements Linux isolés les uns des autres dans des conteneurs partageant le même noyau et une plus ou moins grande partie du système hôte. Le conteneur apporte une virtualisation de l'environnement d'exécution (processeur, mémoire vive, réseau, système de fichier...) et non pas de la machine. Pour cette raison, on parle de « conteneur » et non de « machine virtuelle ».

## 5. Docker

Docker est une solution open source de conteneurs Linux qui s'appuie elle-même sur d'autres composants eux aussi ouverts. Ces briques de base sont en fait communes à tous les types de conteneurs Linux. Initialement, Docker utilisait notamment LXC (Linux Containers) comme implémentation (on parle de driver), mais a ensuite développé sa propre bibliothèque de bas niveau nommée libcontainer pour enfin migrer vers runC, le standard de l'OCI (Open Container Initiative). Ce composant encapsule les fonctionnalités fondamentales, proches du noyau du système d'exploitation, dont la combinaison permet la virtualisation de niveau OS. Aujourd'hui le moteur Docker est construit au-dessus de containerd qui lui-même intègre runC. Pas d'inquiétude néanmoins, la maîtrise de cet empilement de projets et de composants n'est pas fondamentalement utile pour tirer parti de Docker.