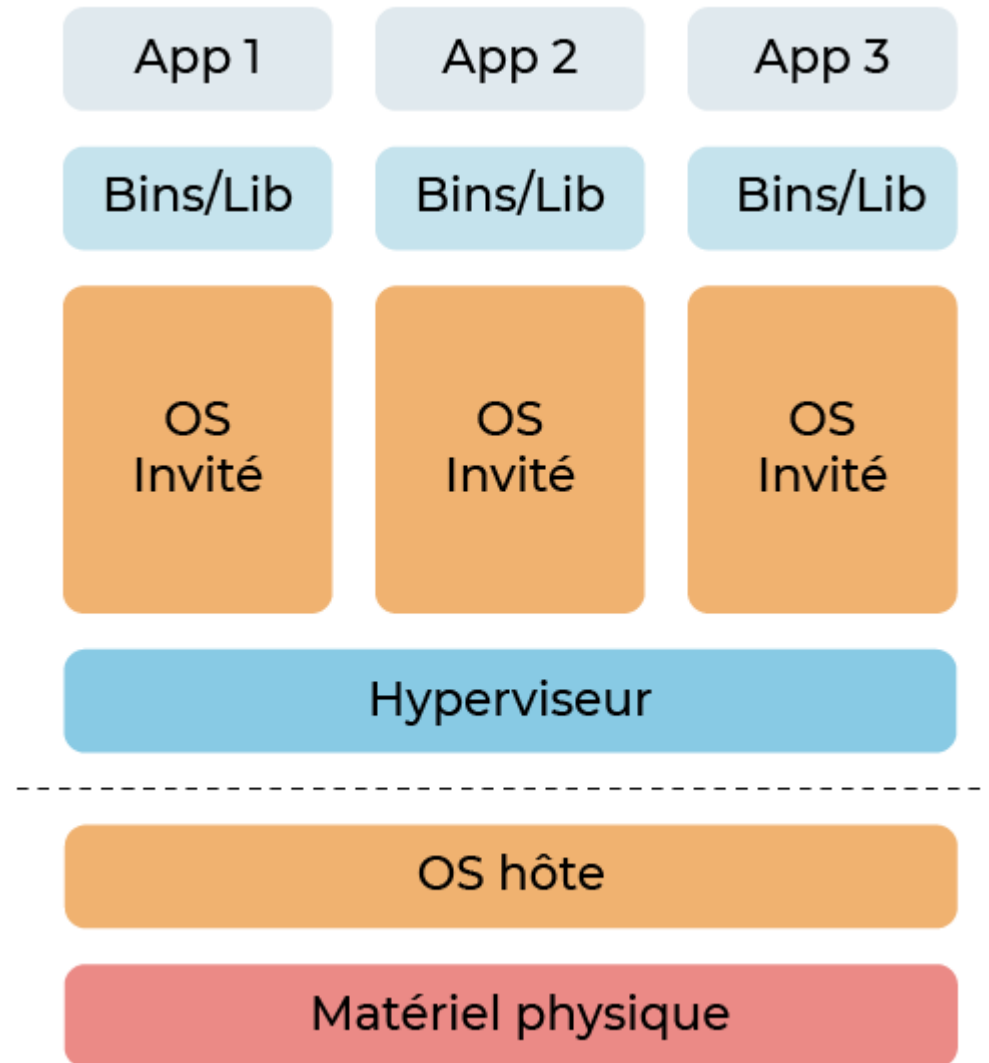


La conteneurisation

Rappels : la machine virtuelle ?

Une machine virtuelle :



Lorsque vous utilisez une machine virtuelle (VM), vous faites ce qu'on appelle de la virtualisation lourde.

- En effet, vous recréez un système complet dans le système hôte, pour qu'il ait ses propres ressources.
- ✓ L'isolation avec le système hôte est donc totale

cependant, cela apporte plusieurs contraintes :

- ✓ une machine virtuelle prend du temps à démarrer ;
- ✓ une machine virtuelle réserve les ressources (CPU/RAM) sur le système hôte.

Mais cette solution présente aussi de nombreux avantages :

- ✓ une machine virtuelle est totalement isolée du système hôte ;
- ✓ les ressources attribuées à une machine virtuelle lui sont totalement réservées ;
- ✓ vous pouvez installer différents OS (Linux, Windows, BSD, etc.).

Mais il arrive très souvent que l'application qu'elle fait tourner ne consomme pas l'ensemble des ressources disponibles sur la machine virtuelle.

- Alors est né un nouveau système de virtualisation plus léger : les conteneurs.

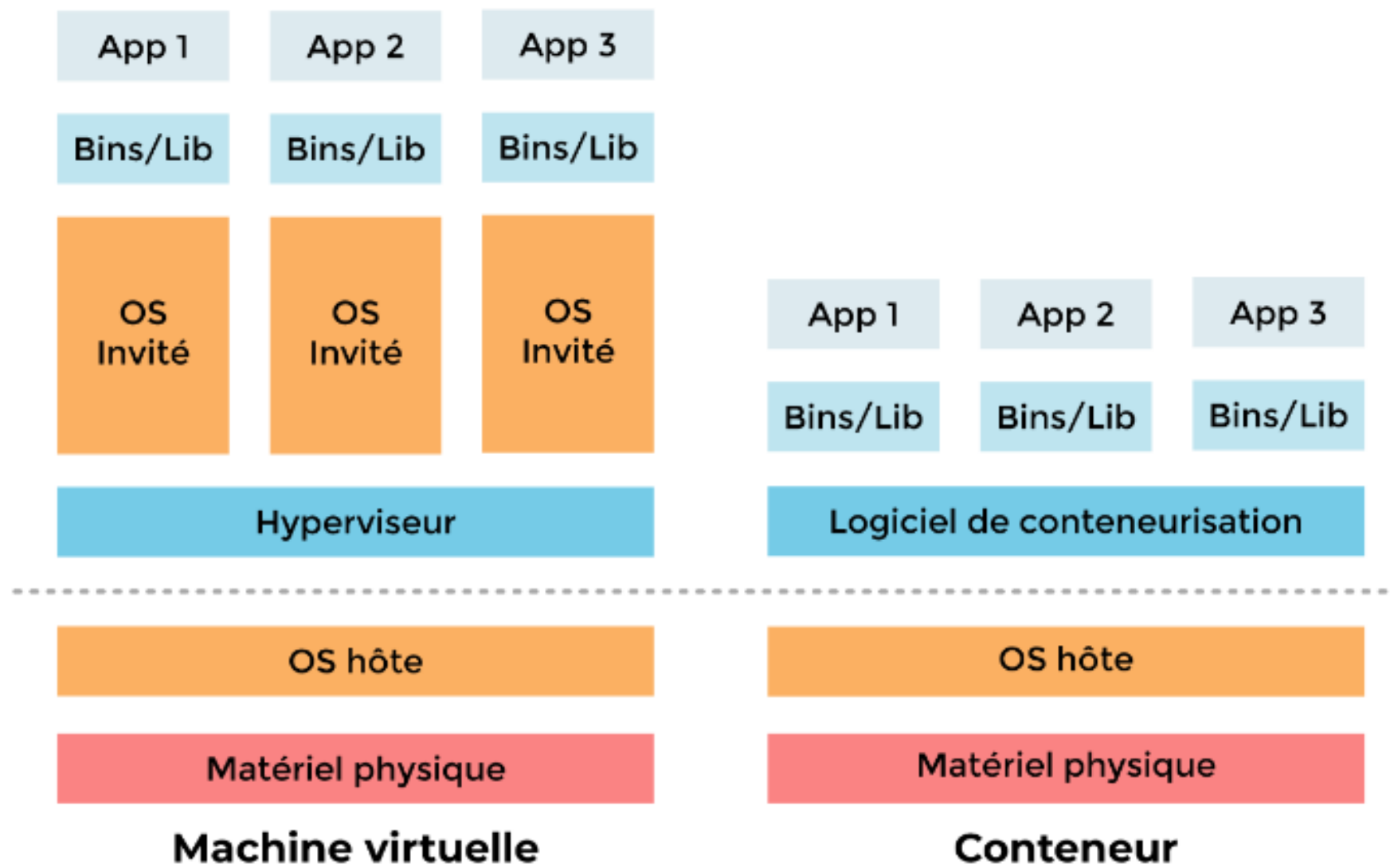
Un conteneur

Un conteneur Linux est :

- ✓ un processus ou un ensemble de processus isolés du reste du système,
- ✓ tout en étant léger.

Le conteneur permet de faire de la virtualisation légère :

- c'est-à-dire qu'il ne virtualise pas les ressources, il ne crée qu'une isolation des processus.
- Le conteneur partage donc les ressources avec le système hôte.



Comparaison entre les conteneurs et les machines virtuelles

Avantages

Ne réservez que les ressources nécessaires

Une des différence importante avec les machines virtuelles :

- un conteneur ne réserve pas la quantité de CPU, RAM et disque attribuée auprès du système hôte.

Ainsi, nous pouvons allouer 16 Go de RAM à notre conteneur, mais si celui-ci n'utilise que 2 Go, le reste ne sera pas verrouillé.

Démarrez rapidement vos conteneurs

Les conteneurs n'ayant pas besoin d'une virtualisation des ressources mais seulement d'une isolation :

- ils peuvent démarrer beaucoup plus rapidement et plus fréquemment qu'une machine virtuelle sur nos serveurs hôtes,

et ainsi réduire encore un peu les frais de l'infrastructure.

Donnez plus d'autonomie aux développeurs

En dehors de la question pécuniaire :

- il y a aussi la possibilité de faire tourner des conteneurs sur le poste des développeurs,

et ainsi de réduire les différences entre la "sainte" production, et l'environnement local sur le poste des développeurs.

Pourquoi utiliser des conteneurs ?

Les conteneurs permettent :

- ✓ de réduire les coûts,
- ✓ d'augmenter la densité de l'infrastructure,
- ✓ tout en améliorant le cycle de déploiement.



Nous retrouvons les conteneurs dans de nombreux contextes. Par exemple, les outils d'intégration continue (CI) de grandes sociétés utilisent des conteneurs. BlaBlaCar utilise des conteneurs pour créer et isoler rapidement des tests/builds d'application.

Les conteneurs étant capables de démarrer très rapidement :

- ils sont souvent utilisés en production pour ajouter des ressources disponibles, et ainsi répondre à des besoins de mise à l'échelle ou de scalabilité.

Mais ils répondent aussi à des besoins de préproduction :

- ✓ en étant légers et rapides au démarrage,

ils permettent de créer des environnements dynamiques et ainsi de répondre à des besoins métier.



Attention, bien que le fonctionnement des conteneurs et des machines virtuelles soient très proches, **l'un ne remplace pas l'autre** ! Pour faire tourner un conteneur, il faut toujours une machine (virtuelle ou non). Ce sont donc deux outils complémentaires à utiliser en connaissance de cause.

En résumé

Vous connaissez maintenant la différence entre conteneur et machine virtuelle, voir les différences entre la virtualisation lourde et la virtualisation légère.

➤ Un conteneur doit être léger,

- ✓ il ne faut pas ajouter de contenu superflu dans celui-ci afin de le démarrer rapidement,
- ✓ mais il apporte une isolation moindre.

À contrario, les machines virtuelles offrent une très bonne isolation,

- ✓ mais elle sont globalement plus lentes
- ✓ et bien plus lourdes.

Questions ??

Prochain chapitre: Docker.