1. **Qu'est-ce que la programmation de conteneurs**

Les conteneurs sont des packages logiciels qui contiennent tous les éléments nécessaires pour s'exécuter dans n'importe quel environnement. Ils peuvent ainsi virtualiser le système d'exploitation et s'exécuter partout : centre de données privé, cloud public, voire ordinateur portable personnel d'un développeur.

1. **Où la programmation de conteneurs est-elle utilisée**

**1. Architecture microservices**

**Les organisations du monde entier adoptent les microservices plutôt que les grandes applications monolithiques pour accélérer le développement et les mises à jour. Les microservices sont des services autonomes qui doivent être déployés et gérés indépendamment. Les conteneurs fournissent à ces microservices des environnements de charge de travail isolés. Ce processus facilite le déploiement et la mise à l'échelle de ces microservices de manière indépendante dans des conteneurs distincts. En outre, les conteneurs facilitent la mise à l'échelle de l'architecture de microservices. Si vous avez plus de visiteurs et vendez plus pendant une saison particulière, vous pouvez facilement augmenter la capacité de votre architecture.**

**2. Déploiement multi-cloud**

**Si vous utilisez des solutions multi-cloud ou hybrides, vous pouvez utiliser des conteneurs pour déployer vos applications dans les différents environnements de cloud. En raison de la nature portable des conteneurs, vous pouvez rapidement les déplacer d'un environnement de cloud à un autre sur site, vers le cloud, ou vice versa. De même, si vous cherchez à migrer d'un environnement sur site vers un environnement en nuage, vous pouvez emballer votre application avec des conteneurs et les déployer ultérieurement sur votre nuage préféré.**

**3. Multi-tenant**

**Il est facile d'héberger une architecture multi-tenant avec des conteneurs. Dans une architecture multi-tenant, une seule instance sert plusieurs clients ; ces clients sont appelés locataires.**

**L'architecture multi-tenant peut également être réalisée par la virtualisation en hébergeant plusieurs machines virtuelles isolées sur un seul serveur, et chaque VM possède un OS invité. (L'architecture multi-tenant sur les conteneurs est légère et plus rapide. Par conséquent, le démarrage du conteneur ne prendra que quelques secondes par rapport aux gigaoctets et aux minutes nécessaires au démarrage d'une VM.**

1. **A quoi sert la programmation de conteneurs**

**Portabilite** : Un conteneur crée un paquet exécutable de logiciel qui est abstrait (non lié ou dépendant) du système d'exploitation hôte, et donc portable et capable de s'exécuter de manière uniforme et cohérente sur n'importe quelle plateforme ou nuage.

**La vitesse** : Les conteneurs sont souvent qualifiés de "légers", ce qui signifie qu'ils partagent le noyau du système d'exploitation (SE) de la machine et ne sont pas encombrés de cette surcharge. Cela permet non seulement d'améliorer l'efficacité des serveurs, mais aussi de réduire les coûts des serveurs et des licences, tout en accélérant les temps de démarrage puisqu'il n'y a pas de système d'exploitation à démarrer.

**Isolation des pannes** : Chaque application conteneurisée est isolée et fonctionne indépendamment des autres. La défaillance d'un conteneur n'affecte pas le fonctionnement continu des autres conteneurs. Les équipes de développement peuvent identifier et corriger tout problème technique dans un conteneur sans que les autres conteneurs ne subissent de temps d'arrêt. En outre, le moteur de conteneur peut tirer parti de toutes les techniques d'isolation de sécurité du système d'exploitation, telles que le contrôle d'accès **SELinux**, pour isoler les défaillances au sein des conteneurs.

**Efficacité** : Les logiciels exécutés dans des environnements conteneurisés partagent le noyau du système d'exploitation de la machine, et les couches d'application d'un conteneur peuvent être partagées entre plusieurs conteneurs.

**Sécurité** : L'isolation des applications en tant que conteneurs empêche intrinsèquement l'invasion d'un code malveillant d'affecter les autres conteneurs ou le système hôte. En outre, des autorisations de sécurité peuvent être définies pour empêcher automatiquement les composants indésirables d'entrer dans les conteneurs ou pour limiter les communications avec les ressources inutiles

1. **Quels sont les avantages de la programmation par conteneur**

**Séparation des responsabilités**

La conteneurisation permet une séparation claire des responsabilités. En effet, les développeurs peuvent se concentrer sur la logique et les dépendances de leur application pendant que les équipes informatiques se consacrent au déploiement et à la gestion, sans avoir à se soucier de détails tels que la version du logiciel ou la configuration spécifique de l'application.

### Portabilité des charges de travail

Les conteneurs peuvent s'exécuter dans pratiquement n'importe quel environnement, ce qui facilite grandement le développement et le déploiement sous Linux, Windows, Mac, sur des machines virtuelles ou des serveurs physiques, sur l'ordinateur d'un développeur, dans un centre de données sur site et, bien entendu, dans le cloud public.

### Isolement des applications

Les conteneurs virtualisent le processeur, la mémoire, l'espace de stockage et les ressources réseau au niveau du système d'exploitation (OS), ce qui offre aux développeurs un aperçu de l'OS logiquement isolé des autres applications

1. **Quels sont les inconvénients de la programmation de conteneurs**

**Assurer la compatibilité multiplateforme**

Le problème majeur est que si une application conçue pour fonctionner dans un conteneur Docker sur Windows, elle ne peut pas fonctionner sur Linux ou vice versa. Cependant, les machines virtuelles ne sont pas soumises à cette limitation.

**Exécuter des applications avec des interfaces graphiques**

En général, les containers sont conçus pour héberger des applications qui s'exécutent en ligne de commande. Bien que nous disposions de quelques moyens (comme la redirection X11) pour permettre l'exécution d'une interface graphique à l'intérieur d'un conteneur Docker, ces moyens sont peu pratiques

**Exécuter des applications aussi rapidement qu'un serveur bare-metal**

Par rapport aux machines virtuelles, les conteneurs Docker ont moins de frais généraux, mais pas zéro. Si nous exécutons une application directement sur un serveur bare-metal, nous obtenons une véritable vitesse bare-metal, même sans utiliser de conteneurs ou de machines virtuelles. Cependant, les conteneurs ne fonctionnent pas à la vitesse bare-metal.

.