

L'automatisation de la création des containers avec un outil de type Docker

Table des matières

I. Outil de type Docker	3
II. Exercice : Quiz	6
III. Automatisation des containers	7
IV. Exercice : Quiz	9
V. Essentiel	10
VI. Auto-évaluation	10
A. Exercice	10
B. Test	11
Solutions des exercices	11

I. Outil de type Docker

Durée : 1 h

Environnement de travail : PC connecté à Internet

Contexte

C'est le nouveau chouchou de l'architecture logicielle. Grâce à la possibilité d'isoler les applications dans des conteneurs virtuels, les architectures applicatives existantes connaissent un nouveau souffle. Dans un monde de micro-services, la danse d'orchestration des conteneurs virtuels a le potentiel de changer la façon dont vous concevez les applications. Les conteneurs ont également changé la façon dont les applications multimédia sont déployées. Ils représentent une solution tangible pour créer les environnements hybrides à forte valeur ajoutée qui sont si populaires dans les entreprises. Ils promettent deux choses : la flexibilité, un développement plus rapide des applications et des mises à jour dans un cycle continu, et l'interopérabilité. Les entreprises utilisent de plus en plus souvent des machines virtuelles (VM). Une machine virtuelle est un système d'exploitation logiciel ou un environnement d'application. Il permet à l'utilisateur de vivre la même expérience que sur un ordinateur physique, avec de nombreux avantages. Plusieurs environnements de systèmes d'exploitation peuvent être exécutés sur un seul ordinateur, en étant isolés les uns des autres. La virtualisation permet également de réduire les coûts organisationnels. Il réduit également la consommation d'énergie. La sauvegarde et la récupération sont également simplifiées. Cependant, les hyperviseurs VM utilisent l'émulation matérielle et nécessitent donc une puissance de calcul élevée. De nombreuses entreprises se tournent vers les conteneurs et, à travers eux, vers Docker pour résoudre ce problème. La conteneurisation est l'une des dernières tendances et l'utilisation de conteneurs dans le développement de logiciels va augmenter de façon spectaculaire, tant en matière d'échelle que de vitesse. Les acteurs du secteur s'attendent à ce que le coût, pour l'instant élevé, de la conteneurisation diminue à mesure que l'environnement évolue et mûrit. L'utilisation des technologies de conteneurisation des applications est très répandue parmi les entreprises et les industries. Elle connaîtra également une croissance rapide dans les années à venir. Plusieurs entités ont déjà commencé à conteneuriser des applications dans le nuage ou à diviser des monolithes existants en conteneurs pour tirer parti de l'architecture de conteneurisation. Dans ce contexte, l'outil type Docker et l'automatisation des containers feront l'objet du développement.

Le terme « *Docker* » fait référence à plusieurs choses : le travail open source de la communauté, les outils basés sur ce travail open source, la société Docker, qui est un soutien majeur de ce travail, et les outils officiellement soutenus par cette société. Le fait qu'une technologie et une entreprise portent le même nom peut prêter à confusion. Docker est une technologie de téléchargement de conteneurs qui vous permet de créer et d'utiliser des conteneurs Linux. La communauté open source Docker s'efforce d'améliorer cette technologie, qui est disponible gratuitement pour tous.

Docker Inc. encourage le travail de la communauté Docker en protégeant sa technologie et en partageant ses succès avec tous les utilisateurs. En conséquence, elle promeut une technologie améliorée et sécurisée pour ses entreprises clientes. Docker utilise le noyau Linux et des fonctionnalités du noyau, telles que les groupes et les espaces de noms pour isoler les processus et les exécuter indépendamment. Il permet à plusieurs processus et applications de fonctionner de manière isolée les uns des autres, en bénéficiant du même niveau de sécurité que les systèmes individuels tout en optimisant l'utilisation de l'infrastructure.

Cela permet de partager facilement un ensemble d'applications ou de services et toutes leurs dépendances entre les environnements. Docker peut également automatiser le déploiement d'une application (ou d'un ensemble de processus liés qui constituent une application) dans un environnement de conteneurs. Ces outils sont construits au-dessus des conteneurs Linux (et sont donc pratiques et uniques), offrant aux utilisateurs un accès sans précédent aux applications, un développement accéléré, un contrôle des versions et des performances.

Complément

L'approche de Docker en matière de conteneurisation repose sur la décomposition des applications, c'est-à-dire sur la possibilité de corriger ou de mettre à jour certaines parties d'une application sans arrêter l'ensemble de celle-ci. En plus de cette approche basée sur les micro-services, Docker permet de distribuer les processus entre différentes applications de manière similaire à une architecture orientée services (SOA). Chaque fichier image Docker est constitué de plusieurs couches. Ces couches sont compilées en une seule image. Chaque fois qu'une image est modifiée, un nouveau calque est créé. Toutes les fois où un utilisateur exécute une commande, telle qu'exécuter ou copier, une nouvelle couche est créée.

Docker utilise à nouveau ces couches pour créer de nouveaux conteneurs, ce qui accélère le processus de construction. Les modifications permanentes sont partagées entre les images afin de maximiser la vitesse, la taille et les performances. La stratification inclut le contrôle de la version. Le journal des modifications est mis à jour à chaque changement, ce qui vous donne un contrôle total sur les images des conteneurs. La régénération est peut-être la caractéristique la plus intéressante de la stratification. Chaque image est constituée de couches. Ainsi, si vous n'êtes pas satisfait de l'itération actuelle d'une image, vous pouvez régénérer la version précédente. Cette fonctionnalité prend en charge le développement agile.

La mise en service, l'approvisionnement et la mise à disposition d'un nouveau dispositif prenaient quelques jours. Il s'agissait d'un processus complexe et long. Aujourd'hui, grâce aux conteneurs Docker, cela peut se faire en quelques secondes. Vous pouvez rapidement mettre des processus similaires à la disposition de nouvelles applications. Et comme il n'est pas nécessaire de redémarrer le système d'exploitation lors de l'ajout ou du déplacement d'un conteneur, le temps de déploiement est encore plus réduit. Et ce n'est pas tout.

La vitesse de déploiement est telle que vous pouvez vous toutefois avoir l'autorisation de créer et de détruire facilement et à peu de frais des données dans des conteneurs. La technologie Docker offre une approche plus granulaire, gérée et basée sur les micro-services qui donnent la priorité aux performances. Docker est une technologie très puissante pour gérer les conteneurs individuels. Cependant, à mesure que le nombre de conteneurs et d'applications conteneurisées (divisées en centaines de composants) augmente, la gestion et l'orchestration deviennent plus complexes.

Finalement, il faut prendre du recul et orchestrer plusieurs conteneurs pour s'assurer que les services (réseau, sécurité, télémétrie, etc.) sont répartis sur tous les conteneurs. C'est là que la technologie Kubernetes vient à la rescousse. Dans un conteneur Docker, vous ne disposez pas des fonctionnalités de type UNIX que les conteneurs Linux traditionnels offrent, ce qui veut dire que vous ne pouvez pas faire usage des processus tels que cron ou syslog avec votre application dans le conteneur. Il y a également des limitations sur le nettoyage des processus enfants lorsque vous en arrêtez un, alors que les conteneurs Linux traditionnels avaient l'habitude de le faire.

Définition

Un conteneur Linux est un processus ou un ensemble de processus isolés du reste du système. Tous les fichiers nécessaires à leur fonctionnement sont disponibles dans une image séparée, ce qui signifie que les conteneurs Linux sont portables et fonctionnent de la même manière. Leur utilisation est donc beaucoup plus rapide que les pratiques de développement qui reposent sur la réplication à partir d'environnements de test traditionnels. Les conteneurs constituent également un élément important de la sécurité informatique puisqu'ils sont reconnus et faciles à utiliser. Supposons que vous développiez une application. Vous travaillez sur un ordinateur portable avec une configuration d'environnement spécifique. D'autres développeurs peuvent travailler sur des ordinateurs ayant une configuration légèrement différente.

L'application que vous développez est basée sur cette configuration. En même temps, votre entreprise dispose d'environnements de développement et de production qui sont standardisés sur la base de leurs propres configurations et des ensembles de fichiers associés. Vous voulez répliquer ces environnements localement autant que possible, mais sans encourir les coûts associés à la réplication des environnements de serveurs. Comment s'assurer que l'application que vous développez fonctionne dans ces environnements, passe le contrôle de qualité et est déployée sans problèmes, retouches ou correctifs ? La réponse est simple : utilisez des conteneurs. Les composants qui constituent une image de conteneur sont similaires à l'installation d'une distribution Linux, car l'image contient tous les RPM, les fichiers de configuration, etc.

Cependant, le déploiement d'images de conteneurs est beaucoup plus facile que l'installation d'une nouvelle version du système d'exploitation. Une fois que vous avez évité la crise, vous pouvez vous mettre au travail. Il s'agit d'un exemple simplifié, mais les conteneurs Linux peuvent également être utilisés pour trouver solution à un bon nombre de problèmes liés à des situations qui exigent portabilité, configurabilité et isolation. Les conteneurs Linux permettent de développer des applications plus rapidement et de répondre aux besoins de l'entreprise dès qu'ils se présentent. Dans certains cas, comme la diffusion de données en temps réel à l'aide d'Apache Kafka, les conteneurs sont essentiels car ils constituent le seul moyen d'assurer l'évolutivité nécessaire aux applications.

Quelle que soit l'infrastructure (sur site, dans le nuage ou hybride), les conteneurs sont parfaitement adaptés. On pourrait donc croire que les conteneurs et la virtualisation sont deux notions identiques, mais ce n'est pas exactement la même chose. Au contraire, les conteneurs et la virtualisation sont complémentaires. Voici une définition simple à des fins de clarification : la virtualisation permet aux systèmes d'exploitation (Windows ou Linux) de fonctionner simultanément sur un seul système matériel. Par exemple : les systèmes ARM Linux utilisent les conteneurs ARM Linux, les systèmes x86 Linux utilisent les conteneurs x86 Linux et les systèmes x86 Windows utilisent les conteneurs x86 Windows. Les conteneurs Linux sont hautement portables, mais doivent être compatibles avec le système sous-jacent.

La virtualisation utilise un hyperviseur pour émuler le matériel afin que plusieurs systèmes d'exploitation puissent fonctionner en parallèle. Il s'agit d'une solution moins légère que les conteneurs. Lorsque vous disposez de ressources limitées et de fonctionnalités restreintes, vos applications doivent être légères et utilisables dans un environnement dense. Les conteneurs Linux fonctionnent sur leur propre système d'exploitation, qu'ils partagent. Cela permet à vos applications et services d'être légers et de s'exécuter rapidement en parallèle. Les conteneurs Linux constituent une nouvelle évolution dans la façon dont les applications sont créées, déployées et gérées. Les images de conteneurs Linux assurent la portabilité et le contrôle de version des applications. Un conteneur Linux nécessite moins de ressources qu'une machine virtuelle.

Il dispose d'une interface standard (démarrage, arrêt, variables d'environnement, etc.), offre une isolation de l'application et peut être plus facilement géré comme un module dans une application plus vaste (conteneurs multiples). En outre, ces applications peuvent être orchestrées à l'aide de plusieurs conteneurs dans différents nuages. Le projet LXC possède une interface de ligne de commande simple qui facilite la prise en main par les nouveaux utilisateurs. Il comprend un environnement de virtualisation au niveau du système d'exploitation qui peut être installé sur de nombreux systèmes basés sur Linux. Vous pouvez y accéder depuis le dépôt de paquets de votre distribution Linux.

Définition

Avant de parler de Docker, il est important de comprendre ce qu'est une image de conteneur. Il s'agit d'un ensemble de processus logiciels légers et indépendants qui regroupent tous les fichiers nécessaires à l'exécution des processus. Ils peuvent exécuter des applications Linux ou Windows. Les conteneurs sont donc similaires aux machines virtuelles, mais présentent un avantage important. Alors que la virtualisation consiste à exécuter bon nombre de systèmes d'exploitation sur un seul système, les conteneurs utilisent un seul noyau de système d'exploitation et isolent les processus applicatifs du reste du système. En termes simples, au lieu de virtualiser le matériel comme un hyperviseur, un conteneur virtualise le système d'exploitation. Il est donc beaucoup plus efficace qu'un hyperviseur en matière de consommation de ressources système. En particulier, il est possible d'exécuter près de 4 à 6 fois plus d'instances d'applications sur un conteneur que sur des machines virtuelles, telles que Xen ou KVM, sur le même matériel. Il s'agit d'une plateforme logicielle open source pour la création, la distribution et la gestion de conteneurs d'applications virtuelles sur un système d'exploitation. Les services ou fonctions d'application et leurs diverses bibliothèques, fichiers de configuration, dépendances et autres composants sont regroupés dans un conteneur. Chaque conteneur en fonctionnement partage les services du système d'exploitation. Docker a été conçu à l'origine pour la plateforme Linux, mais il fonctionne désormais aussi sur d'autres systèmes d'exploitation. La plateforme de conteneurs repose sur sept éléments clés. Docker Engine est un outil client-serveur destiné à soutenir les activités de développement d'applications basées sur des

conteneurs. Le moteur crée un démon de processus serveur qui héberge les images, les conteneurs, les réseaux et les volumes de stockage. Ce démon fournit également une interface client SLI qui permet aux utilisateurs d'interagir avec le démon via l'API de la plateforme.

Le composant Docker Compose est utilisé pour définir la composition des composants dans un conteneur spécifique. Le moteur Docker Swarm prend en charge l'équilibrage de charge par groupe. Cela signifie que les ressources de plusieurs hôtes peuvent être combinées pour agir comme une seule entité. Cela permet aux utilisateurs de mettre rapidement à l'échelle le placement des conteneurs. Il permet de connecter, de créer, d'héberger et de mettre à l'échelle rapidement des conteneurs sur des hôtes Docker. Il offre également un haut degré de portabilité et permet aux utilisateurs d'enregistrer et de partager des conteneurs sur une variété d'hôtes dans des environnements tant publics que privés. Par rapport aux machines virtuelles, Docker offre également un certain nombre d'avantages. Il permet un développement plus efficace des applications en utilisant moins de ressources et un déploiement plus rapide. Les conteneurs sont isolés mais utilisent le même système d'exploitation. Par conséquent, une attaque ou une faille de sécurité du système d'exploitation pourrait compromettre tous les conteneurs. Pour minimiser ce risque, certaines entreprises placent leurs conteneurs dans une machine virtuelle. Docker a subi une faille de sécurité qui a affecté près de 5 % des utilisateurs. Environ 190 000 d'entre eux ont obtenu des données sur les conteneurs après un accès non autorisé à la base de données du Hub. L'agence a demandé aux entreprises et aux particuliers concernés de changer leurs mots de passe.

Exercice : Quiz

[solution n°1 p.13]

Question 1

L'outil Docker est une technologie.

- ☐ Vrai
- ☐ Faux

Question 2

En matière de conteneurisation, l'outil Docker est basé sur la décomposition.

- ☐ Vrai
- ☐ Faux

Question 3

On dispose dans un conteneur de type Docker de la fonctionnalité Unix que dans les conteneurs Linux.

- ☐ Vrai
- ☐ Faux

Question 4

Un conteneur Linux est un ensemble de processus non isolés.

- ☐ Vrai
- ☐ Faux

Question 5

Comparativement à une machine virtuelle, les conteneurs Linux nécessitent plus de ressources.

- ☐ Vrai
- ☐ Faux

III. Automatisation des containers

La définition de Docker montre exactement où se situe cet outil, entre l'automatisation des infrastructures et la virtualisation. En général, Docker est donc équivalent aux logiciels spécialisés dans le déploiement et l'automatisation des infrastructures. Si vous avez utilisé l'un des systèmes d'automatisation, vous avez probablement remarqué que vous avez passé du temps à créer des recettes et d'autres listes de contrôle pour la gestion des serveurs. Cette dépense ne peut être amortie qu'une fois que vous avez un nombre important de serveurs qui doivent être configurés de la même manière. Et de préférence sur un système relativement unifié, car l'engagement en faveur d'une abstraction complète du système d'exploitation sous-jacent rend le maintien des recettes dans le temps plus difficile, voire impossible.

Docker offre un flux de travail très simple. Chaque application possède un ou plusieurs conteneurs nécessaires à son fonctionnement. Chaque conteneur est un dérivé d'un conteneur de base plus simple. En tant que tel, il s'agit d'un modèle en couches, ou, pour les développeurs et les administrateurs qui en comprennent le fonctionnement, d'une lasagne. Chaque conteneur possède un fichier Docker qui répertorie les actions requises pour préparer le conteneur à la production. Ce n'est pas très différent d'écrire un script shell après l'installation, mais c'est beaucoup plus prévisible ! Ces fichiers et les conteneurs qui en résultent sont contrôlés par version, de sorte qu'ils peuvent être gérés comme du code. Vous pouvez également trouver une liste de conteneurs dans la communauté Docker. Docker est construit au-dessus de LXC, il ne gère donc pas de machines virtuelles complètes avec du matériel émulé, mais de simples conteneurs isolés du système hôte par un certain nombre de mécanismes. Bien que seuls les serveurs Linux ayant la même version de noyau doivent être virtualisés, les performances sont garanties en utilisant des conteneurs LXC plutôt qu'une virtualisation complète telle que KVM ou VMware. Les serveurs prennent moins de temps à démarrer que, par exemple, les serveurs et consomment moins de ressources.

Complément

Les images Docker sont compilées à partir de fichiers Docker. Les conteneurs sont créés à partir d'images. En général, un conteneur n'est pas modifié après sa création. En d'autres termes, les conteneurs sont généralement conçus pour être éphémères, ce qui signifie qu'ils peuvent être détruits et remplacés par un nouveau conteneur à tout moment. Si la redondance est suffisante (par exemple, plusieurs serveurs web exécutant le même service), il n'y a pas de mal à détruire le conteneur et à en démarrer un nouveau du même type. Si un élément doit être modifié, comme la version du logiciel ou la configuration, il est d'usage de mettre à jour le fichier Docker et de recréer le conteneur avec la dernière image.

Par conséquent, les conteneurs ne doivent rien contenir qui ne puisse être perdu, et leur restauration doit être une opération extrêmement bon marché. Le schéma Docker Compose ou Swarm est utilisé pour indiquer quels conteneurs composent un environnement particulier et comment ils interagissent entre eux. Ansible et Puppet, quant à eux, sont principalement utilisés pour gérer la configuration de serveurs existants. Ils ne créent pas de nouveaux serveurs, mais les reconfigurent. Par conséquent, Docker et Ansible ont des approches très différentes. Pour cette raison, Ansible et Puppet ne sont pas souvent utilisés pour déployer des conteneurs en production. Néanmoins, leur utilisation conjointe peut présenter des avantages, notamment pour les environnements de développement.

Les conteneurs Docker peuvent être entièrement gérés en mode Docker Compose ou Swarm. C'est généralement une bonne idée. Toutefois, l'utilisation de logiciels d'automatisation tels qu'Ansible ou Puppet présente des avantages. Les conteneurs Docker vous permettent de travailler sans modifier le système hôte, et la création de conteneurs est beaucoup plus rapide qu'avec des machines virtuelles, ce qui rend Docker populaire dans les environnements de développement.

Comme expliqué précédemment, tous les conteneurs peuvent être éphémères et les données importantes peuvent être stockées dans des volumes. Toutefois, cela implique d'ajouter une certaine complexité et d'adapter la philosophie de l'éphémère à une technologie qui n'est pas éphémère par nature (les bases de données). En

outre, cette approche ne plaît pas à de nombreux experts en bases de données. Les mises à jour et les changements de configuration des conteneurs peuvent facilement être déclenchés par un logiciel d'automatisation, ce qui permet de traiter les conteneurs comme des systèmes non continus. Docker n'est parfois utilisé que dans des environnements de développement où des logiciels d'automatisation tels qu'Ansible ou Puppet gèrent les bases de données de production, ce qui peut entraîner une duplication du code. L'utilisation d'un même programme pour gérer les changements de configuration peut réduire les coûts de maintenance. Cela est particulièrement vrai lors de l'écriture de fichiers dans le conteneur lui-même ou lorsque la recreation du chargeur de conteneur prend beaucoup de temps. Essayer de faire des actions non standard avec des fichiers Docker peut être délicat. Par exemple, il est possible d'exécuter deux processus dans un seul conteneur, ce qui peut poser des problèmes car Docker est conçu pour exécuter un seul processus dans un seul conteneur. Dans certains cas, cependant, cela est souhaitable. Par exemple, dans un conteneur PMM exécutant plusieurs processus, il peut être plus facile d'exécuter des processus supplémentaires dans Ansible ou Puppet que dans un fichier Docker.

Exemple

Imaginez un scénario dans lequel un bogue est signalé dans le système de production et est attribué au prochain développeur indépendant disponible. Le développeur examine le rapport de bogue, clone, le dépôt de code source de la version de production de l'application, construit l'application localement et commence à la tester sur son poste de travail de développeur. Notez qu'il n'a pas besoin d'installer localement des outils de développement spécialisés, ni de configurer une version d'outils déjà installés ; il lui suffit d'avoir Docker en fonctionnement sur son poste de travail. Maintenant, il identifie le bogue, modifie le code source, reconstruit et teste à nouveau l'application localement. Ce cycle de récupération peut prendre moins de 10 minutes dans l'environnement de développement Docker, même pour les développeurs qui viennent de rejoindre l'équipe.

Les développeurs peuvent désormais enregistrer leurs modifications et effectuer des demandes de téléchargement en toute confiance. D'un autre côté, si les développeurs doivent configurer leur environnement local avec les versions correctes de tous les outils nécessaires à la version de production actuelle, cela peut prendre des jours sans garantie de correspondance avec la version de production. Pour la plupart des équipes de développement expérimentées, il est évident que l'intégration de Docker dans l'environnement de développement logiciel a beaucoup de sens. Les méthodes traditionnelles de programmation dans un environnement informatique particulier entraînent souvent des erreurs et des échecs lors du transfert du code vers un nouvel emplacement. Lors du déplacement du code d'un ordinateur vers une machine virtuelle, le moteur Docker, avec son approche de packaging universel et ses outils simples pour les développeurs, est devenu la norme industrielle pour la conteneurisation des processus.

Les conteneurs sont souvent qualifiés de légers dans l'industrie. Cela signifie qu'ils partagent le système d'exploitation sous-jacent de l'ordinateur et ne nécessitent pas qu'un système d'exploitation soit associé à chaque logiciel. Les conteneurs sont donc plus petits en capacité native que les machines virtuelles, ont un temps de démarrage plus court et peuvent exécuter plus de conteneurs qu'une seule machine virtuelle sur une seule capacité de calcul.

La conteneurisation permet aux développeurs d'écrire une application une fois et de l'exécuter partout. Cette portabilité est essentielle pour le développement des processus et la compatibilité avec les fournisseurs. Vous pouvez fournir instantanément des applications conteneurisées aux utilisateurs dans l'espace de travail numérique.

La conteneurisation apporte des avantages considérables aux développeurs et aux équipes de développement, qu'il s'agisse d'une flexibilité et d'une portabilité accrues ou d'un meilleur contrôle des coûts. Le conteneur d'applications crée un logiciel exécutable, isolé du système d'exploitation hôte. Il est donc indépendant du système d'exploitation hôte et n'y est pas lié, ce qui le rend portable et capable de fonctionner de manière cohérente et uniforme sur n'importe quelle plateforme ou dans le nuage. Les méthodes d'intégration du système d'exploitation utilisées par les développeurs permettent aussi d'éviter des incohérences telles que l'intégration visant à limiter les fonctionnalités de l'application.

Les développeurs qualifient les conteneurs de « légers » parce qu'ils utilisent le noyau du système d'exploitation hôte et ne sont pas alourdis par des frais généraux supplémentaires. Leur légèreté améliore l'efficacité des serveurs et réduit les coûts des serveurs et des licences. Il réduit également le temps de démarrage car il n'est pas nécessaire de démarrer le système d'exploitation.

Complément

L'utilisation d'un conteneur Docker vous permet de créer une version maîtresse d'une application et de la déployer rapidement à la demande. L'environnement de conteneurs offre une certaine souplesse si vous souhaitez créer à la demande plusieurs nouvelles instances d'applications dans des conteneurs. La technologie des applications conteneurisées offre une excellente évolutivité. Le conteneur d'applications peut gérer des charges de travail croissantes en configurant l'architecture existante pour assurer l'utilisation des ressources grâce à la conception d'applications orientées services. Par ailleurs, un développeur peut ajouter plusieurs conteneurs à un cluster de calcul distribué.

Les logiciels exécutés dans un environnement de conteneurs utilisent le noyau du système d'exploitation hôte. Il faut moins de temps pour les mettre en place et les faire fonctionner, ce qui permet aux développeurs d'exécuter plusieurs conteneurs sur la même puissance de traitement qu'une seule machine virtuelle. Il en résulte une plus grande efficacité du serveur et une réduction des coûts de serveur et de licence. En outre, le moteur de conteneurs peut utiliser des techniques d'isolation de la sécurité du système d'exploitation, telles que le contrôle d'accès SELinux, pour détecter et isoler les défaillances des conteneurs.

L'isolation des applications de conteneur empêche le code malveillant d'avoir un impact sur les autres applications de conteneur ou sur le système hôte. Il est possible de définir des autorisations de sécurité afin de bloquer automatiquement l'accès aux éléments non autorisés qui tentent de se connecter à d'autres conteneurs ou de limiter les communications. L'isolation des applications permet aux développeurs de partager des fonctionnalités supplémentaires sans le facteur risque. Utilisez la plateforme d'orchestration de conteneurs pour automatiser le déploiement, la gestion et la mise à l'échelle des charges de travail et des services de conteneurs. L'orchestration de conteneurs simplifie les tâches de gestion, telles que le déploiement de nouvelles versions d'applications, la mise à l'échelle d'applications conteneurisées ou la fourniture de fonctionnalités de surveillance, de journalisation et de configuration.

Exercice : Quiz

[solution n°2 p.13]

Question 1

L'usage de Docker permet de :

- ☐ Mettre en place une simple application
- ☐ Créer une version maîtresse de l'application

Question 2

Les conteneurs sont :

- ☐ Plus grands que les machines virtuelles
- ☐ Plus petits que les machines virtuelles

Question 3

Avec l'évolution technologique, la conteneurisation a permis de :

- ☐ Développer rapidement les applications
- ☐ Rencontrer des difficultés dans le développement des applications

Question 4

Pour le déploiement et l'automatisation, le Docker est :

- ☐ Équivalent aux logiciels spécialisés
- ☐ Équivalent aux simples logiciels

Question 5

Avec les anciennes méthodes de programmation, on assiste à :

- ☐ Des échecs et erreurs lors du transfert du code
- ☐ La facilitation de déplacement du code vers un autre emplacement

V. Essentiel

Un conteneur Linux est un processus ou un ensemble de processus qui sont isolés du reste du système. Tous les fichiers nécessaires à leur fonctionnement sont disponibles dans une image séparée, ce qui signifie que les conteneurs Linux sont portables et fonctionnent de la même manière. Ils sont donc beaucoup plus rapides à utiliser que les pratiques de développement basées sur la réplique dans les environnements de test traditionnels. Parce que les conteneurs sont populaires et faciles à utiliser, ils constituent également un élément important de la sécurité informatique.

Supposons que vous développez une application. Vous travaillez sur un ordinateur portable avec une certaine configuration d'environnement. D'autres développeurs peuvent travailler sur des ordinateurs ayant des configurations légèrement différentes. L'application que vous développez est basée sur cette configuration et utilise certaines bibliothèques, dépendances et fichiers. Dans le même temps, votre entreprise dispose d'environnements de développement et de production qui sont normalisés en fonction de leurs configurations et des ensembles de fichiers associés. Les conteneurs sont souvent qualifiés de légers dans l'industrie. Par conséquent, les conteneurs ont des performances initiales inférieures à celles des VM, des temps de démarrage plus courts, et il est possible d'exécuter plus de conteneurs sur une seule puissance de calcul que sur une seule VM.

La conteneurisation permet aux développeurs d'écrire une application une fois et de l'exécuter n'importe où. Cette portabilité est essentielle pour le développement des processus et l'interopérabilité des fournisseurs. Les applications conteneurisées peuvent être livrées instantanément aux utilisateurs dans l'espace de travail numérique. L'utilisation de conteneurs apporte des avantages considérables aux développeurs et aux équipes de développement, qu'il s'agisse d'une plus grande flexibilité et portabilité ou d'un meilleur contrôle des coûts. Avec la conteneurisation Docker, vous pouvez créer une version maître d'une application et la déployer rapidement à la demande. La conteneurisation vous donne la possibilité de créer à la demande plusieurs nouvelles copies de votre application dans des conteneurs. La technologie des applications conteneurisées offre une excellente évolutivité. Le conteneur d'applications peut gérer des charges de travail croissantes en reconfigurant l'architecture existante pour garantir une utilisation efficace des ressources grâce à une conception d'applications orientée vers les services. En outre, un développeur peut ajouter plusieurs conteneurs à un cluster de calcul distribué.

VI. Auto-évaluation

A. Exercice

Vous devez expliquer à une personne qui n'a pas de notion sur la conteneurisation de quoi il s'agit.

Question 1

[solution n°3 p.15]

Quelle réponse pouvez-vous donner pour l'éclairer ?

Un développeur vient à vous et vous demande en quoi la conteneurisation peut lui être utile.

Question 2

[solution n°4 p.15]

Quelle réponse allez-vous lui donner ?

B. Test**Exercice 1 : Quiz**

[solution n°5 p.15]

Question 1

Le noyau de l'outil Docker est Linux.

- ☐ Vrai
- ☐ Faux

Question 2

Pour l'hébergement de votre application, le conteneur ne contient pas forcément tous les fichiers.

- ☐ Vrai
- ☐ Faux

Question 3

Le contrôle d'accès est l'une des techniques que le moteur de conteneurs a utilisées.

- ☐ Vrai
- ☐ Faux

Question 4

Depuis des décennies existe le concept de l'isolation des processus de conteneurisation.

- ☐ Vrai
- ☐ Faux

Question 5

Le projet Linux Containers n'est rien d'autre qu'une plateforme de conteneurs.

- ☐ Vrai
- ☐ Faux


Solutions des exercices

Exercice p. 6 Solution n°1**Question 1**

L'outil Docker est une technologie.

☒ Vrai

☐ Faux


 Docker est une technologie de téléchargement de conteneurs qui permet de mettre en place et d'utiliser très facilement des conteneurs Linux.

Question 2

En matière de conteneurisation, l'outil Docker est basé sur la décomposition.

☒ Vrai

☐ Faux


 L'outil Docker, en matière de conteneurisation, est axé sur la décomposition des logiciels ou applications, ce qui veut dire qu'il est possible de corriger ou de mettre à jour certaines portions d'une application sans toutefois arrêter l'ensemble de celle-ci.

Question 3

On dispose dans un conteneur de type Docker de la fonctionnalité Unix que dans les conteneurs Linux.

☐ Vrai

☒ Faux


 Dans un conteneur de type Docker, vous ne disposez pas des fonctionnalités de type UNIX que les conteneurs Linux traditionnels offrent, ce qui signifie que vous ne pouvez pas utiliser des processus comme le syslog avec votre application ou logiciel dans le conteneur.

Question 4

Un conteneur Linux est un ensemble de processus non isolés.

☐ Vrai

☒ Faux


 Un conteneur Linux est un processus, ou du moins un ensemble de processus isolés du reste du système. Les fichiers importants pour leur fonctionnement sont disponibles dans une image dé partagée.

Question 5

Comparativement à une machine virtuelle, les conteneurs Linux nécessitent plus de ressources.

☐ Vrai


☒ Faux

 Un conteneur Linux nécessite beaucoup moins de ressources qu'une machine virtuelle. Ainsi donc, il dispose d'une interface standard, mais aussi il offre une isolation de l'application ou logiciel et peut être plus facilement géré comme un module dans une application plus grande.

Exercice p. 9 Solution n°2


Question 1

L'usage de Docker permet de :

- ☐ Mettre en place une simple application
- ☒ Créer une version maîtresse de l'application
-  L'utilisation d'un conteneur Docker vous permet de mettre en place une version maîtresse d'une application ou logiciel et de la déployer très rapidement à la demande.


Question 2

Les conteneurs sont :

- ☐ Plus grands que les machines virtuelles
- ☒ Plus petits que les machines virtuelles
-  Les conteneurs sont plus petits en capacité native comparativement aux machines virtuelles, qui ont un temps de démarrage plus court mais peuvent exécuter plus de conteneurs qu'une seule machine virtuelle sur une seule capacité de calcul.


Question 3

Avec l'évolution technologique, la conteneurisation a permis de :

- ☒ Développer rapidement les applications
- ☐ Rencontrer des difficultés dans le développement des applications
-  La conteneurisation a facilité et permis aux développeurs la mise en place des logiciels rapidement et de façon plus sûre.


Question 4

Pour le déploiement et l'automatisation, le Docker est :

- ☒ Équivalent aux logiciels spécialisés
- ☐ Équivalent aux simples logiciels
-  L'outil Docker est équivalent aux logiciels qui sont spécialisés dans le déploiement et l'automatisation des infrastructures. Si vous avez utilisé l'un des systèmes d'automatisation, vous avez probablement remarqué que vous avez passé beaucoup de travail et de temps à créer des recettes.

Question 5

Avec les anciennes méthodes de programmation, on assiste à :

- ☒ Des échecs et erreurs lors du transfert du code
- ☐ La facilitation de déplacement du code vers un autre emplacement
-  Les méthodes traditionnelles ou anciennes de programmation dans un environnement informatique particulier conduisent très souvent à des erreurs et des échecs lors du transfert du code informatique vers un nouvel emplacement.

p. 10 Solution n°3

Il s'agit d'un type de virtualisation du système d'exploitation dans lequel les applications s'exécutent dans des espaces utilisateurs. Un conteneur d'applications est un environnement informatique entièrement packagé et portable. Les conteneurs contiennent tout ce qui est nécessaire pour exécuter une application, y compris les binaires, les bibliothèques, les fichiers de dépendance et les fichiers de configuration, le tout encapsulé et isolé.

p. 10 Solution n°4


Les conteneurs sont un système convivial pour les développeurs car ils leur permettent d'utiliser des environnements de développement et de production différents, ce qui constitue un obstacle courant au développement d'applications web. Une équipe de développement peut écrire une application sur un ordinateur portable Windows, mais elle ne fonctionnera pas sur une station de travail Mac. Avec la conteneurisation, l'équipe crée localement la même image que celle qui fonctionne en production. Combinées au bon flux de travail, les applications conteneurisées peuvent minimiser les cas où une application fonctionne bien dans un endroit mais échoue lorsqu'elle est exécutée dans un autre.

Exercice p. 11 Solution n°5**Question 1**

Le noyau de l'outil Docker est Linux.

☒ Vrai

☐ Faux


 L'outil Docker utilise le noyau Linux et des fonctionnalités du noyau comme les groupes et les espaces de noms pour isoler ou écarter le processus et les exécuter de façon indépendante.

Question 2

Pour l'hébergement de votre application, le conteneur ne contient pas forcément tous les fichiers.

☐ Vrai

☒ Faux


 Il faut comprendre que le conteneur héberge le logiciel ou application qui prend en compte les bibliothèques. Ce qui veut dire qu'il peut être transporté vers la production sans effets secondaires.

Question 3

Le contrôle d'accès est l'une des techniques que le moteur de conteneurs a utilisées.

☒ Vrai

☐ Faux


 Le moteur de conteneurs peut faire usage des techniques d'isolation de la sécurité du système d'exploitation. À titre illustratif, le contrôle d'accès SELinux sert à détecter et isoler les défaillances des conteneurs.

Question 4

Depuis des décennies existe le concept de l'isolation des processus de conteneurisation.

☒ Vrai

☐ Faux


 Bien évidemment, le concept d'isolation des processus et de conteneurisation existe depuis des dizaines d'années. Toutefois, l'introduction des outils open source Docker a fait progresser l'adoption de la technologie des applications qui sont conteneurisées.

Question 5

Le projet Linux Containers n'est rien d'autre qu'une plateforme de conteneurs.

☒ Vrai

☐ Faux

 Le projet Linux Containers n'est rien d'autre qu'une plateforme de conteneurs qui a des sources ouvertes. Ces sources fournissent un ensemble d'outils.