**ENTRAINEUR DEVOPS**

Voici quelques prérequis qui peuvent être utiles pour travailler en tant que développeur DevOps :

1. Connaissance approfondie d'au moins un langage de programmation de haut niveau, comme Python, Java, C# ou Ruby.
2. Bonnes compétences en bash et en script.
3. Connaissance des systèmes de contrôle de version, tels que Git.
4. Connaissance de l'infrastructure et de la configuration de serveur, ainsi que de la virtualisation.
5. Bonne compréhension de la gestion de projet et des processus de développement logiciel.
6. Connaissance des outils et des technologies de l'écosystème DevOps, tels que Docker, Ansible, Jenkins et Kubernetes.
7. Bonnes compétences en communication et en travail d'équipe.

Il est important de noter que cette liste n'est pas exhaustive et que les prérequis peuvent varier en fonction de l'emploi spécifique et de l'industrie.

Haut du formulaire

Bas du formulaire

**Docker**

Docker est un outil de conteneurisation qui permet de créer, de réussir et d'exécuter des applications dans des conteneurs isolés. Les conteneurs sont des packages légers qui incluent tout ce dont une application a besoin pour fonctionner, y compris le code, les bibliothèques, les dépendances et les fichiers de configuration.

Voici quelques avantages de l'utilisation de Docker :

1. Portabilité : les conteneurs Docker peuvent être déployés sur n'importe quel serveur ou cloud qui exécute Docker, ce qui rend le déploiement d'applications plus facile.
2. Isolation : les conteneurs Docker sont isolés les uns des autres et du système d'exploitation hôte, ce qui permet de s'assurer que les applications fonctionnent de manière prévisible, quelle que soit l'environnement de déploiement.
3. Gestion des dépendances : les conteneurs Docker inclut toutes les dépendances nécessaires à l'application, ce qui facilite la gestion des dépendances et évite les problèmes de version.
4. Rapidité : les conteneurs Docker sont rapides à démarrer et à arrêter, ce qui en fait un choix idéal pour les applications qui doivent être redémarrées souvent, comme celles qui sont mises à jour fréquemment.

Docker est largement utilisé dans le monde DevOps pour automatiser le déploiement et la gestion d'applications.

**Voici les étapes de base pour utiliser Docker** :

1. Installez Docker sur votre ordinateur en suivant les instructions fournies sur le site de Docker ( [https://www.docker.com/](https://www.docker.com/" \t "_new) ).
2. Ouvrez une invitation de commandes et saisissez la commande "docker version" pour vérifier que Docker est correctement installé et fonctionne.
3. Tapez la commande "docker run hello-world" pour effectuer le conteneur "hello-world" et vérifiez que Docker fonctionne correctement.
4. Pour effectuer votre propre conteneur, vous devez d'abord créer une image de conteneur à partir de votre code. Vous pouvez le faire en demandant un fichier "Dockerfile" qui décrit les étapes de construction de l'image de conteneur.
5. Utilisez la commande "docker build" pour construire l'image de conteneur à partir de votre Dockerfile.
6. Utilisez la commande "docker run" pour exécuter le conteneur à partir de l'image de conteneur que vous avez créée.

Voici quelques exemples de commandes Docker courantes :

* "docker run" : exécute un conteneur à partir d'une image de conteneur.
* "docker start" : démarrer un conteneur qui a été arrêté.
* "docker stop" : arrête un conteneur qui est en cours d'exécution.
* "docker build" : intégré une image de conteneur à partir d'un Dockerfile.
* "docker pull" : télécharge une image de conteneur à partir d'un registre.
* "docker push" envoie : une image de conteneur vers un registre.

Il existe de nombreuses autres commandes et options disponibles pour gérer et utiliser Docker. Pour en savoir plus, consultez la documentation de Docker ( <https://docs.docker.com/> ) ou faites une recherche en ligne.

Haut du formulaire

Haut du formulaire

Bas du formulaire