

TP devOps

Etape 1 - Questions Théoriques sur le DevOps (5 points)

Qu'est-ce que le DevOps et comment améliore-t-il le processus de développement logiciel ?

Expliquez le concept d'Intégration Continue (CI) et de Déploiement Continu (CD).

Quel est le rôle des tests automatisés dans le DevOps ?

Comment Docker facilite-t-il le DevOps ?

Quelle est la différence entre un conteneur et une machine virtuelle ?

Pourquoi utiliser un système de gestion de version comme Git dans un environnement DevOps ?

Expliquez le concept d'Infrastructure as Code (IaC) et donnez un exemple.

Quels sont les avantages du monitoring et du logging dans le DevOps ?

Comment la culture DevOps influence-t-elle la collaboration entre les équipes de développement et d'opérations ?

Quels sont les défis communs rencontrés lors de l'adoption du DevOps et comment les surmonter ?

Étape 2 -Partie Pratique - Configuration de l'Infrastructure as Code

Dockerfile pour le Projet PHP :

Le Dockerfile fourni est conçu pour une application PHP s'exécutant avec Apache. Il installe également l'extension mysqli, nécessaire pour les interactions avec MySQL, et active le module rewrite d'Apache, utile pour la réécriture d'URL.

Configuration de docker-compose.yml

La configuration docker-compose.yml permet de déployer votre application dans un conteneur Docker, en exposant le port 80 pour un accès via un navigateur web. Elle monte également le répertoire courant dans le conteneur pour faciliter le développement et le test en reflétant les modifications du code source en temps réel.

Configuration de .gitlab-ci.yml

Le fichier .gitlab-ci.yml définit deux étapes de base pour le pipeline CI/CD : test et deploy. La partie test vérifie la syntaxe du fichier PHP principal (vous devrez ajuster le chemin si nécessaire). La partie deploy contient un script de déploiement fictif que vous devrez remplacer par vos commandes de déploiement réelles, en fonction de votre environnement cible.

Adaptation aux Besoins Spécifiques

Tests supplémentaires : Pour les tests, envisagez d'ajouter des scripts spécifiques à votre application, comme des tests unitaires avec PHPUnit ou des tests fonctionnels.

Sécurité et performance : N'oubliez pas de configurer des mesures de sécurité adéquates pour votre application et d'optimiser les performances des conteneurs.

Dépendances : Si votre application nécessite des extensions PHP supplémentaires ou d'autres services (comme une base de données), vous devrez étendre ces fichiers de configuration pour inclure ces éléments.