

Introduction à Docker Dans un Contexte DevOps

1. Introduction :

L'objectif de ce TP est de fournir une introduction pratique à Docker pour les débutants en ingénierie DevOps. Docker est une technologie populaire utilisée pour la conteneurisation des applications, offrant une approche portable et efficace pour le déploiement et la gestion d'applications.

a. Prérequis :

- Connaissance de base des concepts de virtualisation et de développement d'applications.
- Un environnement de développement prêt à être utilisé avec Docker installé.

b. Objectifs :

- Comprendre les concepts de base de Docker.
- Apprendre à créer, exécuter et gérer des conteneurs Docker.
- Comprendre les avantages et les cas d'utilisation de Docker dans un contexte DevOps.

c. Durée estimée de la première partie : 3 heures

2. Introduction à Docker

- Installation de Docker sur votre environnement de développement.
- Création d'un Compte Docker Hub

3. Utilisation de Docker

- Création d'un conteneur Docker à partir d'une image existante : nginx port 8090
- Test et validation du fonctionnement du conteneur nginx
- Exploration du contenu du Conteneur
- Création d'un deuxième Conteneur nginx sur le port 8091
- Gestion des conteneurs Docker : liste, démarrage, arrêt, suppression.

4. Exécution d'une application SpringBoot dans un conteneur Docker.

- Créer une Application SpringBoot avec un seul Contrôleur qui affiche le message « Bonjour »
- Ecrire le Dockerfile pour créer une image de cette application
- Exécuter cette image dans un Conteneur
- Compréhension des couches d'image et de la réutilisabilité des images.

5. Compréhension de l'Application SpringBoot

- Comprendre l'application SpringBoot Fournie
- Installer MongoDB et Créer les Bases et Collections Nécessaires

- Tester l'Application SpringBoot Fournie et vérifier qu'elle accède à la Base Correctement en utilisant JUnit et Postman.

6. Containerisation de l'Application

- Lancer un container avec l'image officielle mongo
- Lancer le container de l'application SpringBoot fournie et vérifier qu'ils communiquent ensemble
- Explorer le contenu des deux containers pour vérifier l'absence d'exceptions
- Faites communiquer les deux containers et vérifier le bon fonctionnement de l'application avec Postman
- Lancer cette application sur des ports différents

7. Docker Compose

- Ecrire un Fichier DockerCompose pour faciliter le lancement de l'application
- Exécuter l'application en utilisant DockerCompose
- Quelle est l'utilité de DockerCompose ?

8. Intégration de Docker dans un environnement DevOps

- Etudier et Expliquer rôle de Docker dans un flux de développement et de déploiement continu.
- Discuter sur les bonnes pratiques et les considérations de sécurité lors de l'utilisation de Docker en production.

9. Création de premiers Pipeline avec Jenkins

- Créer un container avec l'image jenkins (juste pour savoir qu'on peut soit installer jenkins soit utiliser une image)
- Créer un Pipeline permettant de :
 - o Faire un Pull à partir de GitHub du code d'une application au choix
 - o Créer une image de cette application
 - o Push de cette image vers le DockerHub
 - o Schématiser ce Pipeline
- Créer un deuxième impliquant Docker, Ansible, GitHub, Maven et d'autres outils selon vos propositions. Expliquer son rôle et donner son schéma

Remarques :

- Pour faciliter la compréhension et l'interaction, des démonstrations en direct peuvent être organisées pour chaque étape.
- Vous êtes encouragés à poser des questions et à discuter des problèmes rencontrés tout au long du TP.