07 février 2022

Kerem COSKUN

[Nom de la société]

[Adresse de la société]

Objectifs Consignes j1

[Sous-titre du document]

1. Commandes pour gérer les conteneurs et les images

Docker permet créer, déployer et exécuter des conteneurs de manière efficace. Un [conteneur](https://www.freecodecamp.org/news/a-beginner-friendly-introduction-to-containers-vms-and-docker-79a9e3e119b/?gi=ab8480e95fe3) enveloppe l’application d’un logiciel dans une boîte invisible avec tout ce dont il a besoin pour s’exécuter. Ils sont surtout légers, portables et permettent aux développeurs de créer, déployer et exécuter efficacement des applications distribuées.

Pour créer un conteneur il faut taper la commande suivante :



Ici, nous créons un conteneur qui utilisera le port 80 ayant pour nom webhost nginx

Nous avons plusieurs commandes qui permettent de manipuler ces conteneurs :

La commande Docker PS qui permet de lister les conteneurs qui sont en train de tourner.



Docker container start [argument] ou l’argument est le nom du conteneur permet de démarrer ton conteneur.



Docker container stop [argument] permet de d’arrêter ton conteneur.



Docker rm [argument] supprime le conteneur serene\_hypatia.



Docker rmi [Argument] permet de supprimer une image.



1. Variables d’environnement



1. Monitoring des conteneurs.

Docker top [argument] permet d’afficher les processus en cours du conteneur indiqué

Une image contenant texte, bleu

Description générée automatiquement

Docker inspect [argument] est une commande permet d’avoir les informations bruts du docker, on peut notamment y trouver les paramètres réseaux.

Une image contenant texte

Description générée automatiquement

Docker logs [argument] permet de voir les journaux de connexions et d’exécutions du conteneur indiqué.

Une image contenant texte

Description générée automatiquement

Docker exec -it [Argument] /bin/bash permet d’accéder à la commande shell du conteneur.



Net-tools  
iputils-ping