Compte-rendu du TD part 01- Découverte de Docker

Lors de ce TD, nous avons exploré les bases de Docker, un outil puissant pour la gestion de conteneurs. Nous avons couvert les principaux concepts, ainsi que les étapes essentielles pour travailler avec Docker.

1. Prérequis et installation :

Nous avons commencé par vérifier les prérequis pour ce tutoriel, qui ne nécessite aucune compétence spécifique en dehors d'une familiarité de base avec la ligne de commande et l'utilisation d'un éditeur de texte. Nous avons également vérifié l'installation de Docker sur notre système, en utilisant le guide de démarrage de Docker, qui propose des instructions détaillées pour différentes plates-formes (Mac, Linux, Windows). Il est essentiel de s'assurer que Docker est correctement installé avant de passer à l'étape suivante.

2. Exécution du premier conteneur :

Nous avons ensuite exécuté notre premier conteneur en utilisant l'image "Alpine Linux", une distribution légère de Linux. La commande "docker pull" nous a permis de télécharger l'image depuis le registre Docker, et "docker images" nous a permis de voir la liste des images sur notre système. En utilisant "docker run", nous avons lancé un conteneur basé sur l'image Alpine Linux et avons exécuté des commandes à l'intérieur de ce conteneur. Nous avons également exploré la commande "docker ps" pour afficher les conteneurs actuellement en cours d'exécution.

3. Terminologie Docker :

Pour clarifier les termes couramment utilisés dans l'écosystème Docker, nous avons discuté des concepts tels que les images (système de fichiers et configuration de l'application), les conteneurs (instances en cours d'exécution d'images Docker), le démon Docker (service en arrière-plan gérant les conteneurs), le client Docker (outil en ligne de commande permettant d'interagir avec le démon), et le Docker Store (registre d'images Docker)

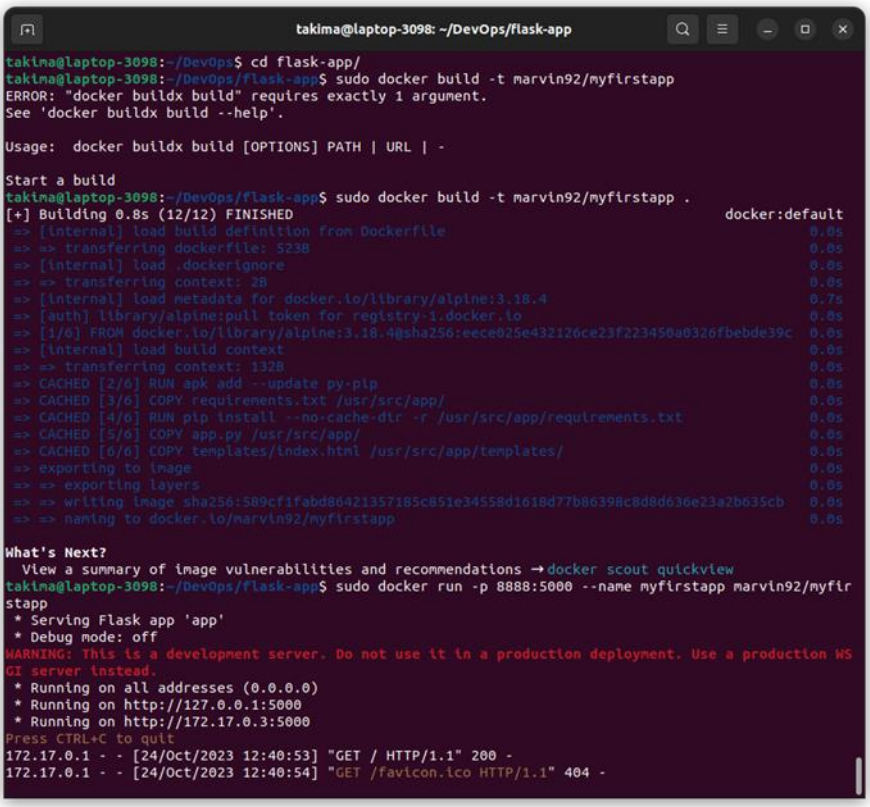
4. Déploiement d'applications Web avec Docker :

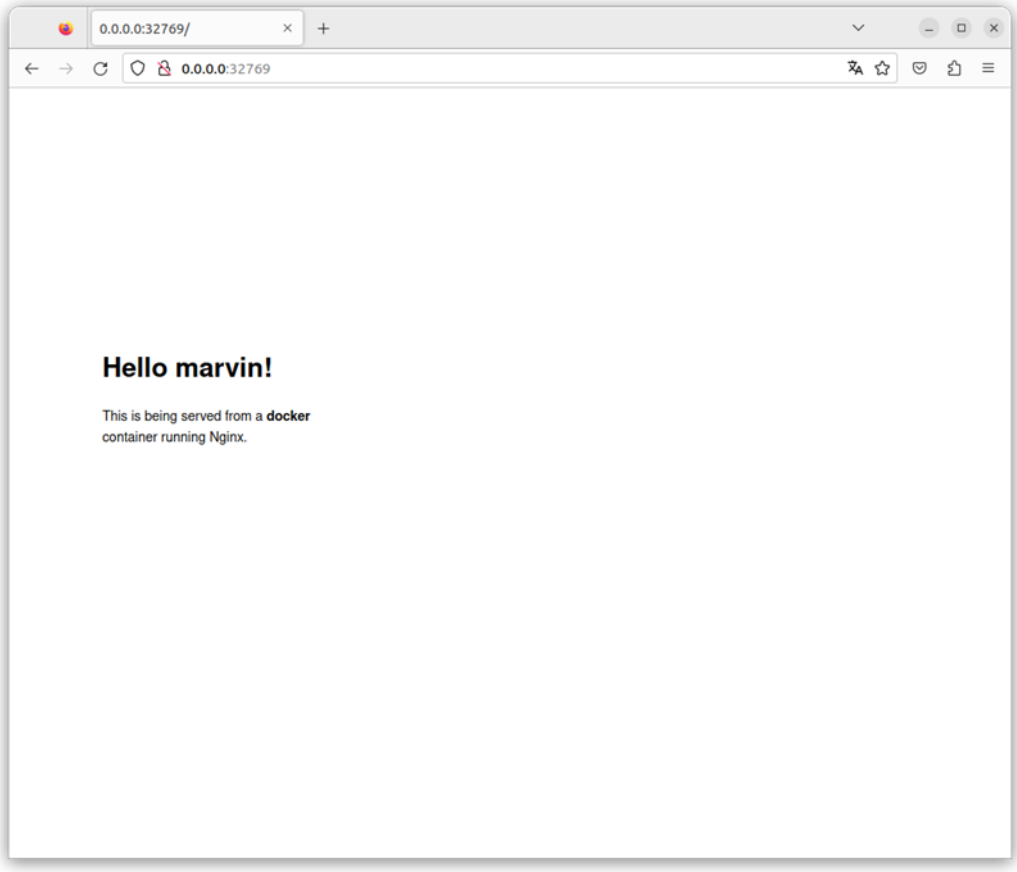
Après avoir acquis une compréhension de base de Docker, nous avons exploré le déploiement d'applications web avec Docker. Nous avons utilisé un exemple d'un site web statique basé sur une image existante, "dockersamples/static-site". Nous avons appris comment exécuter un conteneur, le configurer pour afficher un site web, et l'accéder via un navigateur. De plus, nous avons discuté de la création d'images Docker personnalisées.

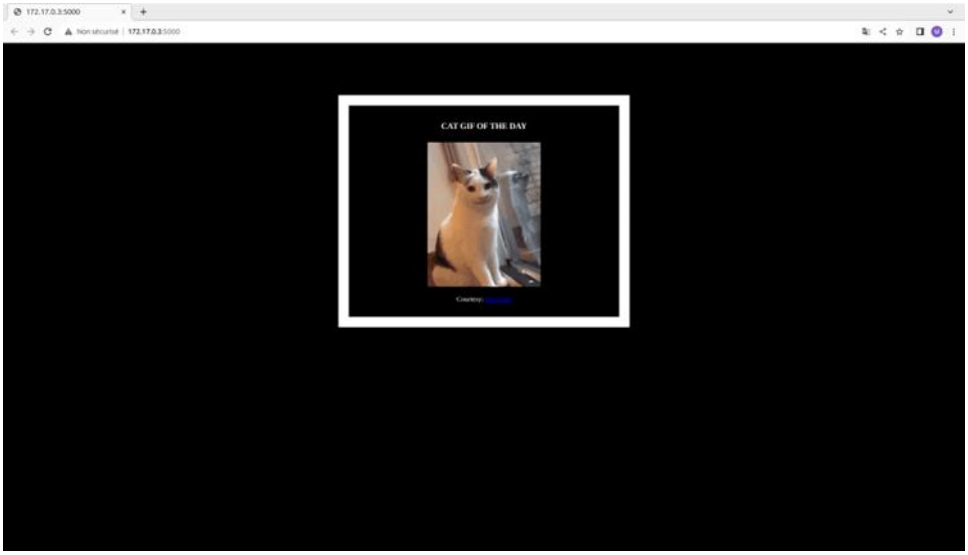
```
takima@laptop-3098:~/DevOps$ sudo docker run --name static-site -e AUTHOR="marvin" -d -P dockersamples/static-site
13baca32b14e51b9cd06c8bad738028de753a21419e5af16079b1b7ebf3e4539
takima@laptop-3098:~/DevOps$ sudo docker port static-site
80/tcp -> 0.0.0.0:32769
80/tcp -> [::]:32769
443/tcp -> 0.0.0.0:32768
443/tcp -> [::]:32768
```

5. Création de notre propre image Docker :

La dernière partie du TD nous a guidés dans la création d'une image Docker personnalisée. Pour cela, nous avons développé une petite application web Flask qui affiche des images de chats aléatoires. Nous avons écrit un Dockerfile pour définir les étapes de création de l'image, y compris l'installation des dépendances Python et la copie des fichiers de notre application dans le conteneur. Enfin, nous avons construit et exécuté notre image personnalisée, obtenant ainsi une application web Flask fonctionnelle.







Ce TD nous a fourni une introduction complète à Docker, allant des concepts de base à la création d'images personnalisées. Nous avons appris comment tirer parti de la flexibilité et de l'efficacité des conteneurs pour le déploiement d'applications. Docker est un outil essentiel pour les développeurs et les administrateurs système, et ce tutoriel nous a permis de commencer à explorer son potentiel.