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.



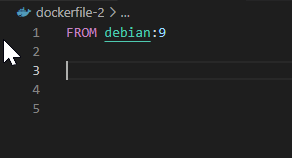
1. Création d’un dockerfile :

Pour créer un dockerfile il suffit d’ouvrir un dossier sur Visual Studio Code et cliquer de créer un fichier et le nommer dockerfile

Une image contenant texte

Description générée automatiquement

Une fois celui-ci créer, commencez par indiquer l’image et la version que vous voulez utiliser.



Une fois ceci fait, sauvegardez à l’aide des raccourcis « CTRL + S »

Vous verrez une petite balène bleu à côté de votre nom de fichier comme sur le screen ci-dessus.

1. Création d’un volume :

Nous avons créé le volume volume-test dans le dossier data de l’image ubuntu.



Pour inspecter le volume créer faire la commande suivante :

Une image contenant texte

Description générée automatiquement

Le volume volume-test a bien été créé dans le dossier data d’ubuntu.

1. Sauvegarder dockerfile sur dockerhub

Pour sauvegarder le dockerfile, il faudrait que nous créions notre image personnalisée en lançant la commande docker build -t [lenomdetonimage] [choisir le dockerfile].

Une image contenant texte

Description générée automatiquement

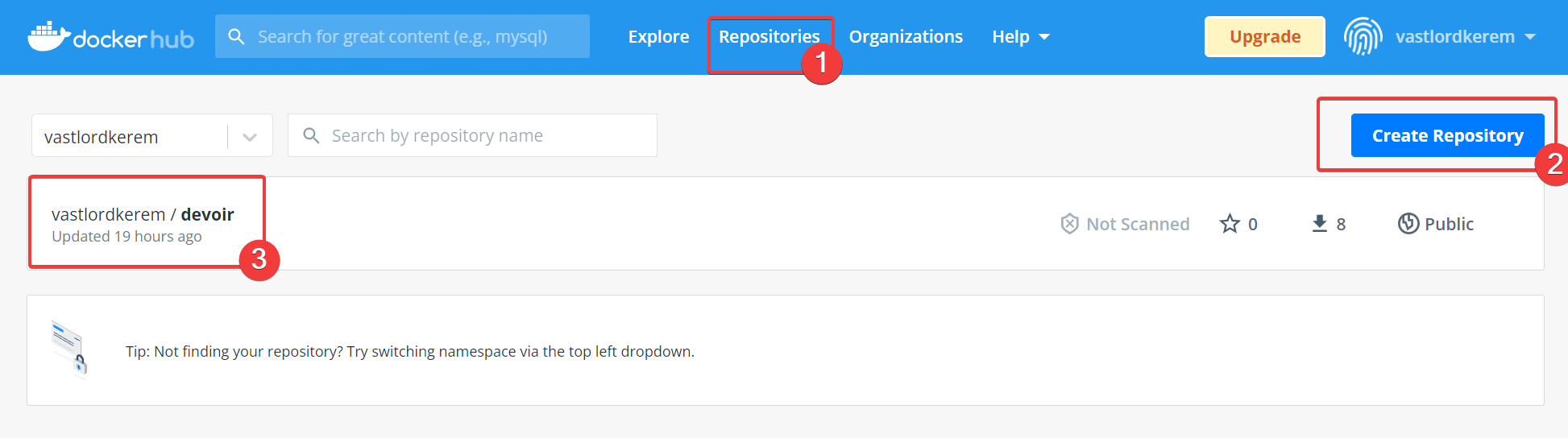
Une fois ceci fait lorsque l’on fait « docker images » nous constatons que notre image est créée.

Une image contenant texte

Description générée automatiquement

Il faudrait ensuite créer un compte sur docker hub puis créer un répertoire.

1. Cliquer sur Repositories
2. Créer le répertoire
3. Le chemin pour déposer le répertoire ici sera vastlordkerem/devoir



Une fois ceci fait, il faudrait créer le tag avec la commande suivante :

Le dockerfile :latest sera le nom de notre depot et latest le tag puis vastlordkerem/devoir sera le chemin du dépôt avec le tag latest.



Puis on le pousse avec la commande suivante.



Une fois ceci fait, nous devrions avoir dans notre dépôt de docker Hub ceci :

Une image contenant texte

Description générée automatiquement

1. Pull notre image personnalisé

On va essayer de pull notre image publié sur docker hub, notre chemin est « vastlordkerem/devoir »

En fesant « docker images » nous constatons que nous n’avons pas encore l’image.

Une image contenant texte

Description générée automatiquement

On pull l’image du docker hub en tapant la commande suivante :

Une image contenant texte

Description générée automatiquement

Le fait qu’il indique Already exists est tout à fait normal, on peut voir que le status est OK.

Lorsque nous faisons un « docker images » de nouveau, nous constatons que l’image est bien là

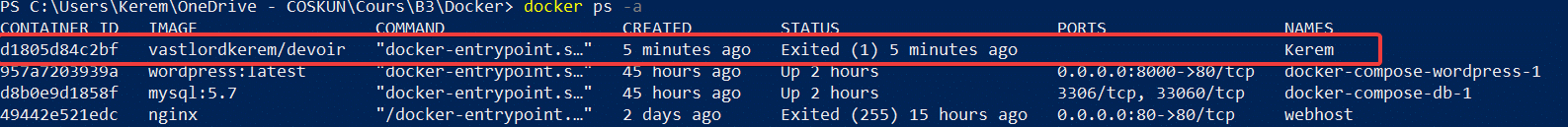
Une image contenant texte

Description générée automatiquement

Nous allons maintenant créer notre conteneur à partir de notre image personnalisée en faisant la commande suivante.



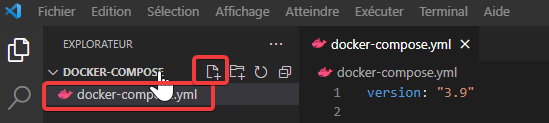
Une fois avoir fait cette manipulation, nous constatons que notre conteneur a bien été créer à partir de notre image vastlordkerem/devoir.



1. Docker-compose

Docker-Compose est un outil qui permet d’associer plusieurs conteneurs ainsi que des paramètres pour y créer toute une configuration personnalisée.

Il suffit de créer une fichier suivi de « Docker-compose.yml » l’extention étant très importante, vous y verrez une petite baleine rose.



Ainsi, nous avons créer notre docker-compose il nous faut maintenant y associer les paramètres.

Une image contenant texte

Description générée automatiquement