

DOCKER – EXPLOITER UN CONTENEUR

Exercice 13 – Script python

Pour cet exercice, nous allons avoir besoin de télécharger le script python fournit « comptage.py », et de le lancer depuis un conteneur construit par une image officielle Python. Ce script analysera une phrase que vous lui passerez en paramètre et vous renverra le nombre de voyelle que celle-ci comporte. Tout un programme !

1. Télécharger le script python « comptage.py » des documents liés au cours
 2. Lancer un conteneur basé sur une image Python avec une consigne infinie (il doit tourner en permanence)
-

Exercice 13 – docker cp

3. Copier votre fichier comptage.py à l'intérieur de votre conteneur. Pour ça, utilisez la commande suivante depuis le dossier où se trouve votre fichier :

```
docker cp <nom du fichier> <id du conteneur>:/< nom du fichier >
```

Ici, vous demandez d'effectuer une copie de votre fichier local et de la placer dans votre conteneur

Exercice 13 – docker exec

4. Enfin, lancer une commande permettant de déclencher le script que vous venez de copier dans le conteneur au moyen de la commande :

```
docker exec -it <id> python /<nom du fichier> <Phrase de test>
```

Ici, vous demander d'exécuter une commande dans votre conteneur « python <nom du fichier> <phrase de test> », et de récupérer le message grâce à l'option `-it`. Cela veut dire que l'interpréteur python sera lancé, le script testera la phrase passée en paramètre et vous donnera le nombre de voyelle qu'elle comporte !

Exercice 14 – Script PHP

Pour cet exercice, je vous fournis cette fois un script PHP rédigé en PHP 7.4 « primechecker.php ». En révision de tous ce que nous avons vu jusque-là, vous allez chercher une image PHP 7.4, monter un conteneur basé sur cette image, récupérer et y copier le script PHP, et le lancer avec la commande « docker exec ».

Ce nouveau script est différent. Il attend un nombre en paramètre (et plus une phrase entre guillemets, juste un nombre sans guillemet). Il vous dira, à la sortie, si celui-ci est un nombre premier, et si ce n'est pas le cas, quel est le nombre premier inférieur le plus proche.

Exercice 15 – docker stats

Afin d'observer les ressources prises par les conteneurs actifs, utilisez la commande :

`docker stats`

Celle-ci permet d'afficher en temps réel des infos de consommation CPU, mémoire (et limite), échanges réseaux et avec le matériel, et le nombre de process simultanés. Cela permet de bien surveiller en cas de baisse de performance qui peut être responsable. Pour quitter la vue des stats, appuyez simplement Ctr + C

Exercice 16 – Exploration

Allez rechercher, via Docker Hub, ou via un navigateur de recherche, des images docker de démo amusante et/ou utiles à votre quotidien (jeux, logiciel ou progiciel, langage de programmation, frameworks web, etc.).

Par exemple, j'ai trouvé pour vous une image publique d'un snake game dans votre terminal, écrit en Go, « dockerisé » et publié sur Docker Hub :

```
docker run -ti dyego/snake-game
```

A vous de rechercher et partager vos images docker amusantes et/ou utiles !

Exercice 17 – Nettoyage

Plusieurs commandes permettent de faire le ménage sur votre poste de travail.

- Supprimer tous les conteneurs non lancés :

`docker container prune`

- Supprimer toutes les images non utilisées :

`docker image prune`

- Supprimer toutes les ressources Docker inutilisées (**commande dangereuse**) :

`docker system prune -a`



Récapitulatif

Prenez maintenant le temps de noter toutes les commandes nouvellement apprises, ce qu'elles font voire les options que vous pouvez passer en paramètre. Cette prise de notes vous sera utile !