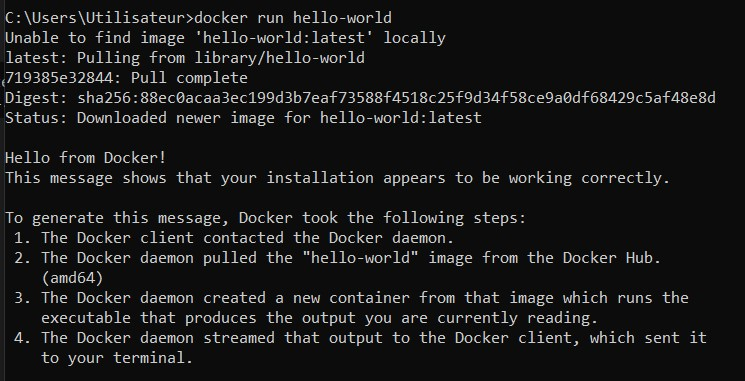
TD de Docker :

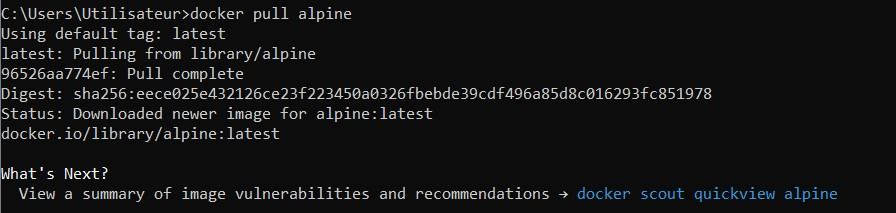
-Permet de vérifier si le DOCKER est véritablement installé



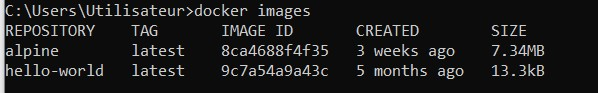
-Permet d’exécuter un conteneur de test :



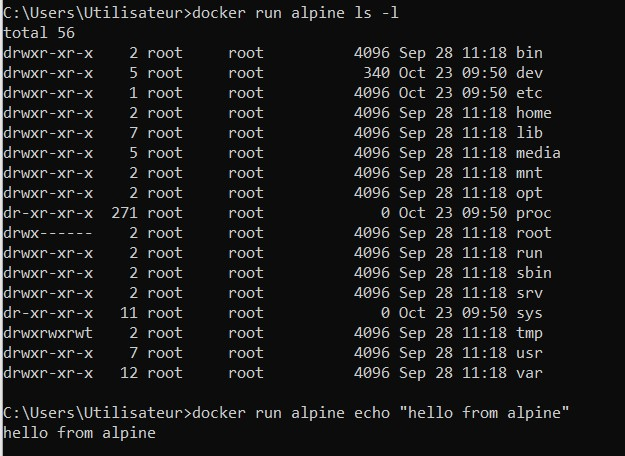
-Ici nous configurons « Alpine Linux container »



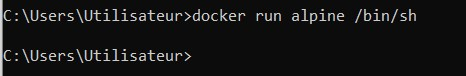
-On utilise « docker images » pour voir la liste des images dans notre système.



-Ici, nous exécutons un conteneur docker basé sur cette image



-Ici ce n’est pas une erreur car ces shells interactifs se fermeront apres l’exécution de commandes scriptées , a moins qu’ils ne soient exécutées dans un terminal interactifs.

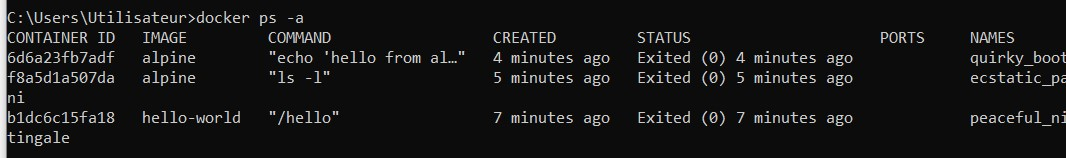


Donc que pour cet exemple ne se termine pas nous devons exécuter « docker run -it alpine /bin/sh »

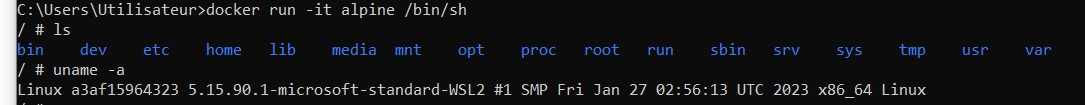
- « docker ps » permet de voir tous les conteneurs en cours d’exécution.



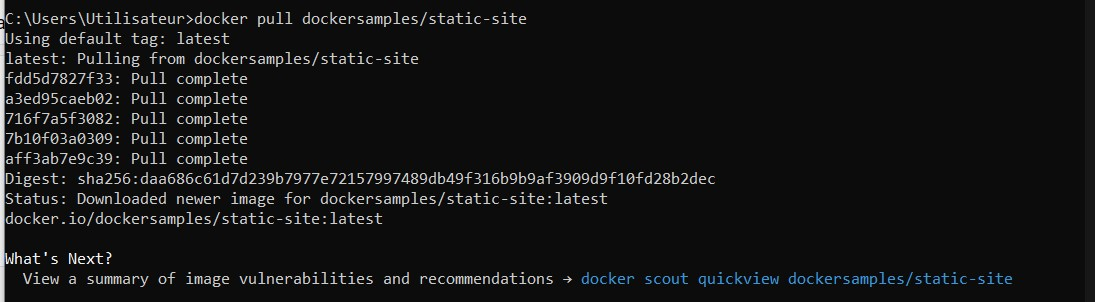
-« docker ps –a » permet de voir toutes les containers qu’on a déjà exécuté.



-Avec la commande « docker run –it alpine /bin/sh » nous pouvons exécuter plusieurs commandes dans un conteneur.



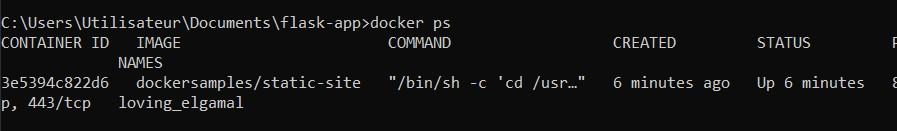
-L’image que nous avons téléchargé vient d’un site web d’une seule page qui a déjà été créé pour cette démo et qui est également disponible sur la [dockersamples/static-site](https://hub.docker.com/r/dockersamples/static-site).



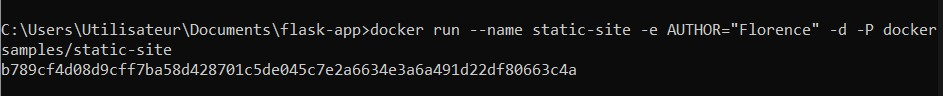
-Comme l’image n’existe pas sur votre hôte docker, le démon Docker le récupère d’abord dans le registre puis l’exécute en tant conteneur.



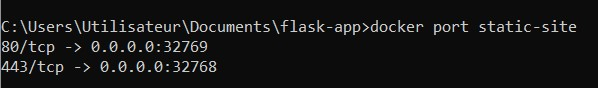
-On a exécuté la commande « docker ps » pour voir les conteneurs qui sont en train de s’exécuter.



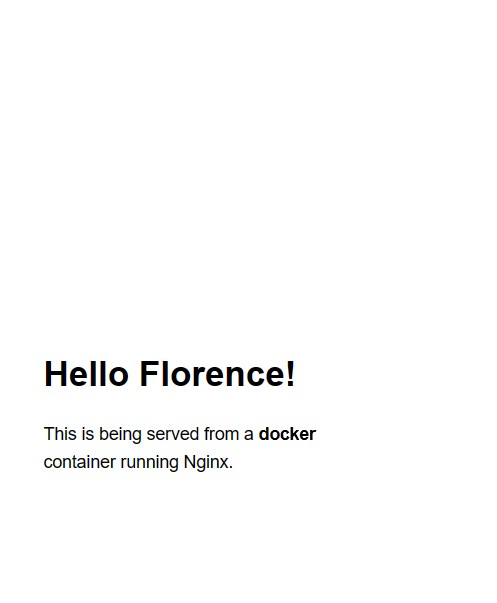
-Maintenant lançons un conteneur en mode détaché comme indique ci –dessous :



-On peut maintenant voir les ports en exécutant la commande docker port

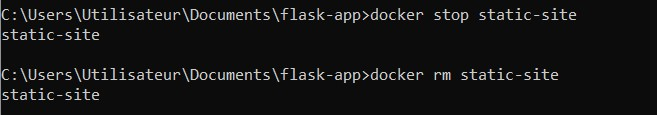


-On exécute sur windows avec [localhost:32769](http://localhost:32769/)



-

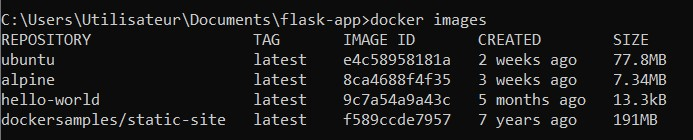
-Ici, on arrête et on enlève le conteneur.



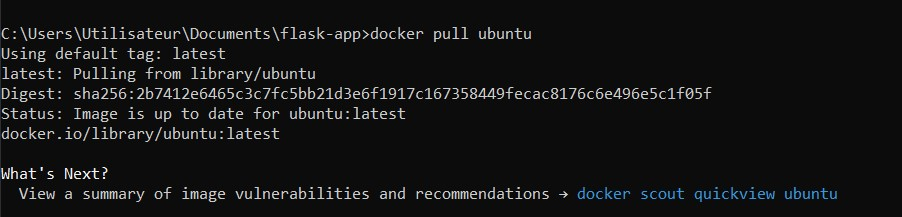
-On constate que les conteneurs ont bien disparu.



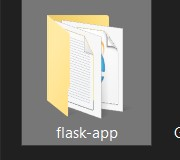
-On affiche les conteneurs qui s’exécutent :



-Permet d’extraire une image :



-Création du répertoire flask-app



-Création des fichiers : app.py, exigences.txt, templates/index.html, Dockerfile (Fichier docker)

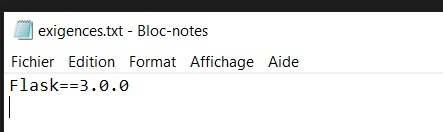
Qu’on mettra dans le répertoire flask-app.



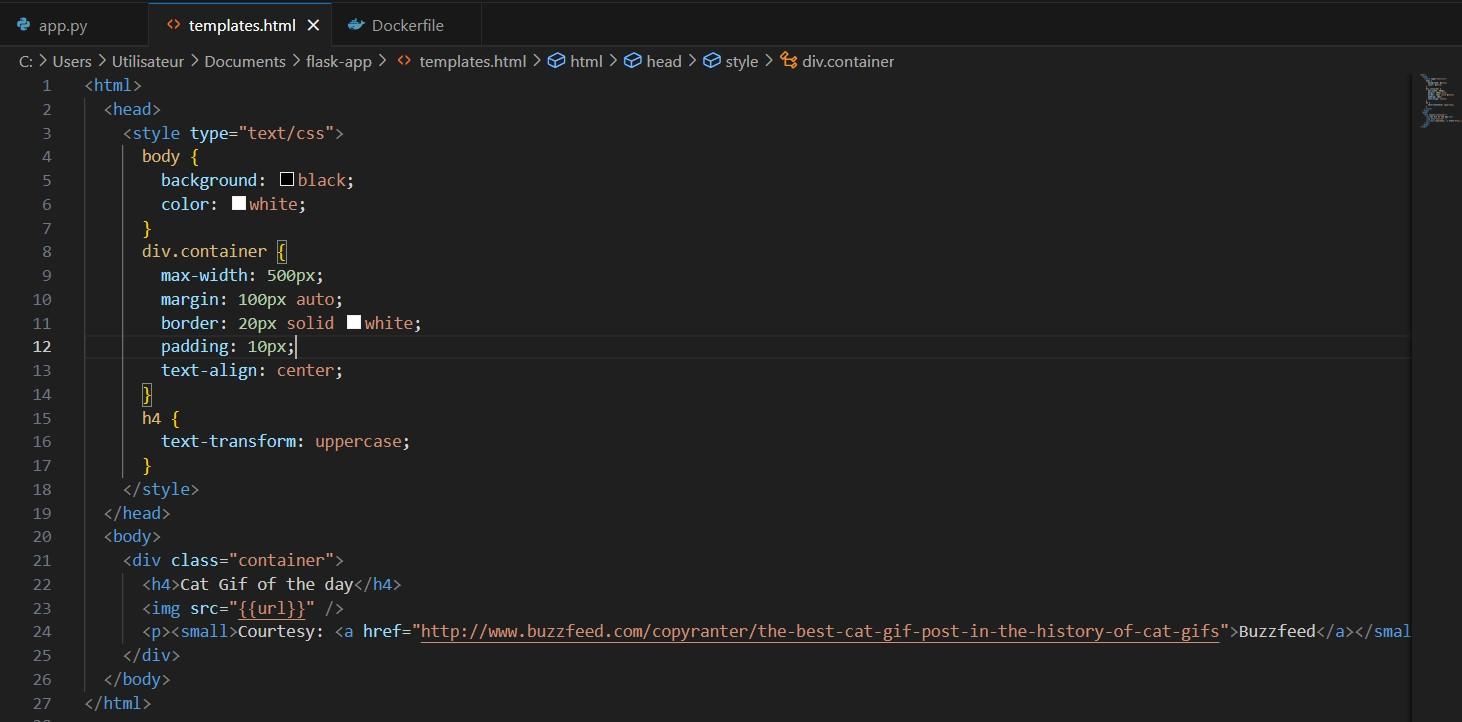
-Contenu de app.py



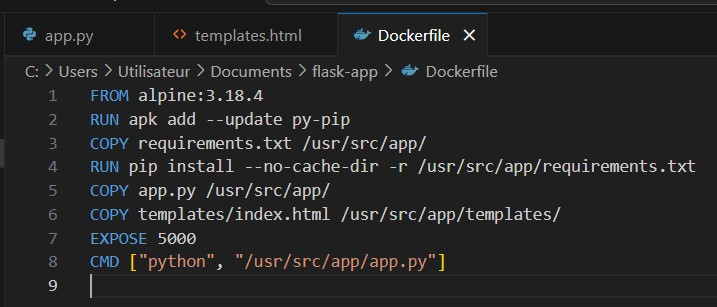
-Contenue de exigences.txt



-Contenue de Templates/index.html :



-Contenue de Dockerfile



-

