**Docker – Python Data**

Ce document est un rapport contenant toutes les informations liées au TP de Python Data. Il donnera toutes les finormations nécéssaires à la comprehension de Docker.

# Introduction

Docker est un outil open source permettant de faciliter le lancement d’applications via la création de conteneurs. Ainsi, il permet au développeur de pouvoir exécuter des applications depuis n’importe quel serveur sans se soucier du système d’exploitation de la machine hôte permettant ainsi une plus grand flexibilité et portabilité vise à vis d’une application. Les conteneurs créer via le processus dit de conteneurisation sont construits sur des capacités du noyau linux.

L'objectif d'un conteneur est le même que pour un serveur dédié virtuel : héberger des services sur un même serveur physique tout en les isolant les uns des autres. Un conteneur est cependant moins figé qu'une machine virtuelle en matière de taille de disque et de ressources allouées.Un conteneur permet d'isoler chaque service : le serveur web, la base de données, une application peuvent être exécutés de façon indépendante dans leur conteneur dédié, contenant uniquement les dépendances nécessaires.

# Premier Conteneur docker

Tout d’abord, pour installer Docker, il suffit de se rendre sur le lien suivant :

<https://www.docker.com/get-started>

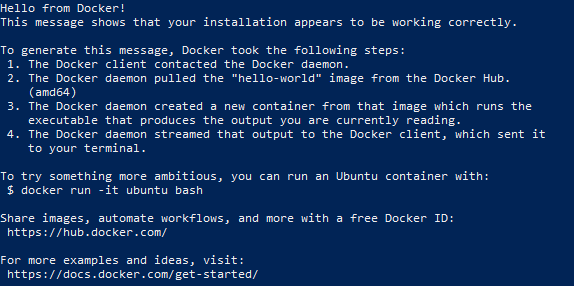
Une fois docker installé, nous allons exécuter notre premier container Windows. Pour cela, il suffit juste d’effectuer la commande suivante dans le répertoire de votre choix :

*docker run -it ubuntu bash*

Rajouter à la commande :

*--detach (-d)* pour pas que le container s’arrête.

Le résultat suivant est censé apparaitre :



Annexe 1 : Résultat affichage docker run

Pour lancer notre docker compose, on va exécuter la commande suivante :

*docker-compose up -d*

Une suite d’instruction va se lancer, une image va se créer et sera alors visible via la commande suivante :

*Docker images*

# Commande

|  |  |
| --- | --- |
| **Commande** | **Description** |
| **docker ps** | Informations concernant les processus docker |
| **Docker images** | Informations concernant l’image docker |
| **docker stop $(docker ps -a -q)** | Arrêter tout les conteneurs |
| **docker rm $(docker ps -a -q)** | Supprimer tous les conteneurs |
| **docker stop NomDuConteneur** | Pour arrêter le conteneur |
| **Docker rm NomDuConteneur** | Supprimer le conteneur |
| **Docker rmi NomDuConteneur** | Supprimer l’image |