**Compréhension de la problématique (point de vue Docker):**

Docker est une plateforme de conteneurisation, probablement la plus populaire et la plus utilisée. L a virtualisation à l'aide des hyperviseurs a permit la possiblité de lancement de plusieurs OS sur une meme machine, en les isolant les uns des autres, toutefois les hyperviseurs de ces VMs requiert une émulation du hardware ainsi une quantité de ressources importante, une solution serait d'exploiter les conteneurs et plus exactement: les Dockers.

**Fonctionnement de Docker :**

Le principe de Docker repose sur celui des images container; qui est un ensemble de processus logiciels léger et indépendant, regroupant tous les fichiers nécessaires à l’exécution des processus : code, runtime, outils système, bibliothèque et paramètres. Ils peuvent être utilisés pour exécuter des applications Linux ou Windows. Les containers sont donc proches des machines virtuelles, mais présentent un avantage important. Alors que la virtualisation consiste à exécuter de nombreux systèmes d’exploitation sur un seul et même système, les containers se partagent le même noyau de système d’exploitation et isolent les processus de l’application du reste du système. Pour faire simple, plutôt que de virtualiser le hardware comme l’hyperviseur, le container virtualise le système d’exploitation. Il est donc nettement plus efficient qu’un hyperviseur en termes de consommation des ressources système. Concrètement, il est possible d’exécuter près de 4 à 6 fois plus d’instances d’applications avec un container qu’avec des machines virtuelles comme Xen ou KVM sur le même hardware.

Docker est une plateforme logicielle open source qui permet de créer, déployer et de gérer des containers d'applications virtualisées dur un système d'exploitation. Les services ou fonctions de l’application et ses différentes bibliothèques, fichiers de configuration, dépendances et autres composants sont regroupés au sein du container. Chaque container exécuté partage les services du système d’exploitation.

**Les composants principaux de Docker :**

a. Docker Engine : outil client-serveur sur lequel repose la technologie de container pour prendre en charge les tâches de création d’applications basées container. Le moteur crée un processus daemon server-side permettant d’héberger les images, les containers, les réseaux et les volumes de stockage. Ce daemon fournit aussi une interface SLI client-side permettant aux utilisateurs d’interagir avec le daemon via l’API de la plateforme.

b. Dockerfiles : les containers créés

c. Docker Hub : un outil SaaS permettant aux utilisateurs de publier et de partager des applications basées container via une bibliothèque commune

d. Docker Swarm : C'est un mode du Docker Engine qui permet de faire le LoadBalancing pout l'équilibrage de charge

Différence entre image et conteneur : un conteneur est une instance d'exécution d'une image plus précisément un conteneur est ce que l'image devient en mémoire lorsqu'elle est exécutée (c'est-à-dire une image avec un état ou un processus utilisateur).

Outils Dockers et processus générale d'implémentation

**Avantages et inconvénients de Dockers**

**++** Composer, créer, déployer et échelonner rapidement des containers sur les hôtes Docker

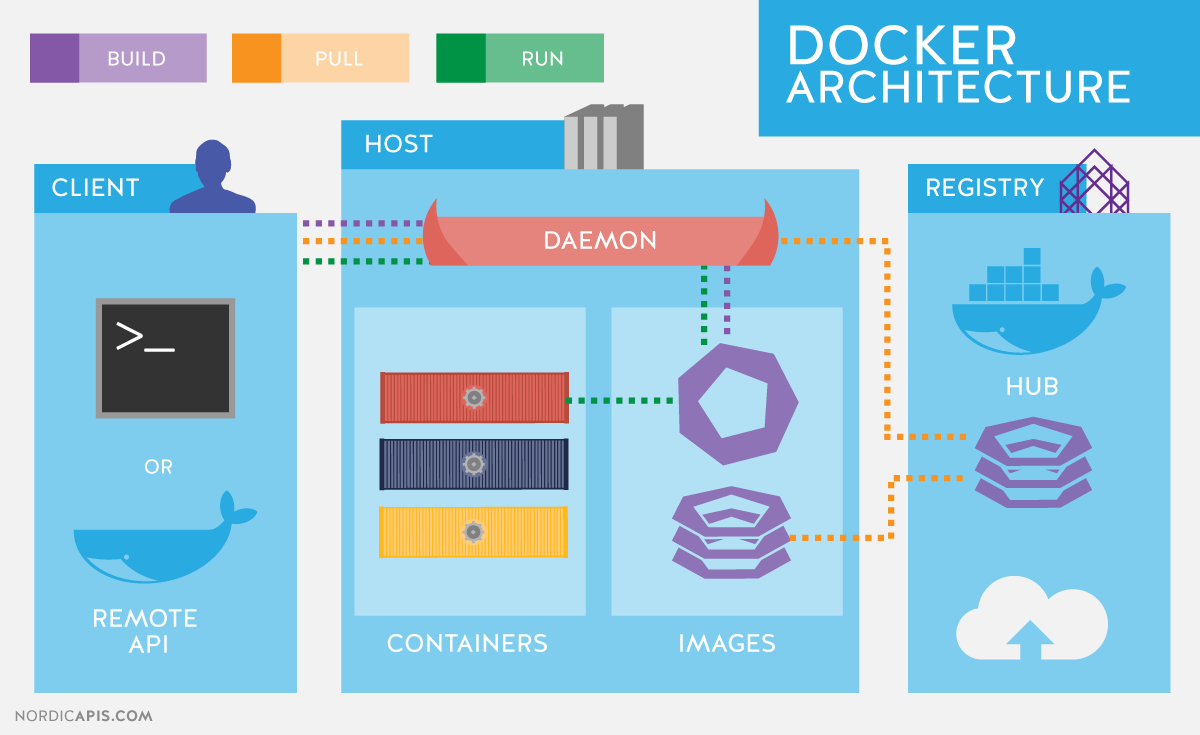
**++** un haut degré de portabilité, ce qui permet aux utilisateurs de s’enregistrer et de partager des containers sur une large variété d’hôtes au sein d’environnements publics et privés

**++** développer des applications de façon plus efficiente, en utilisant moins de ressources, et de déployer ces applications plus rapidement

**- -** difficulté de gérer de façon efficiente un grand nombre de containers simultanément (Ce qui nous pousse à considérer des outils d'orchestration)

**- -** Les containers sont isolés, mais partagent le même système d’exploitation. De fait, une attaque ou une faille de sécurité sur l’OS peut compromettre tous les containers (pour remédier à ce problème, on peut exécuter les dockers sur une machine virtuelle isolée)

**Architecture Docker:**

****

Réf : <https://roygeagea.github.io/docker-tutorials/Architecture/docker-architecture.png>

**Couches virtualisées :**

****

Réf : https://vmarena.com/wp-content/uploads/2018/08/DOCK02.png

**Alternatives et concurrants de Docker**

a. CoreOS rkt

b. Cononical LXD

c. Virtuzzo

d. OpenVZ

**Etude de l'existant par rapport à Docker pour Bash**

<https://github.com/tianon/docker-bash>

<https://github.com/89luca89/distrobox> (many linux distros in one app !)

<https://github.com/M-J-Hooper/BishBashBosh> (multi user bash)

**Etude de l'existant par rapport à Docker pour Packet Tracer**

<https://github.com/search?q=docker+packet+tracer&type=Repositories>