Bases de données documentaires et distribuées, http://b3d.bdpedia.fr

A propos de Docker

L'objectif

Nous allons travailler sur des systèmes de gestion de données **distribués**.

Un système distribué fonctionne, de manière coordonnée, sur plusieurs machines, ou **serveurs**.

Dans ce préliminaire au cours, nous allons étudier Docker, un outil qui permet de simuler un système distribué, avec très peu de ressources, sur un seul ordinateur (par exemple votre portable).

Votre objectif : maîtriser l'installation d'un système avec Docker

Vous devez pouvoir installer en quelques clics CouchDB, MongoDB, Cassandra, Elastic-Search et bien d'autres.

Système distribué : rappels

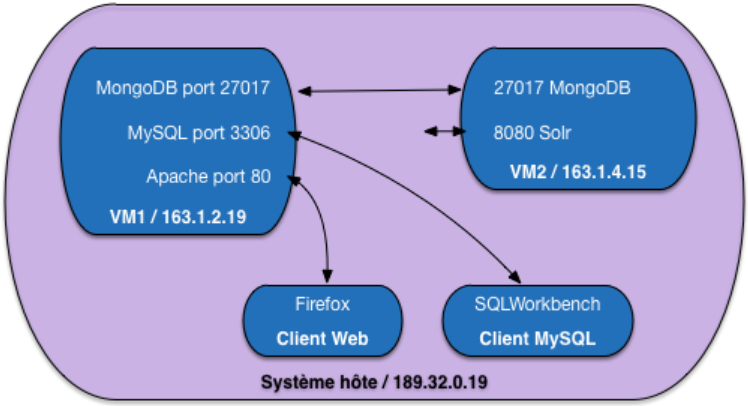
- un **serveur machine** est un ordinateur, tournant sous un système d'exploitation, et connecté en permanence au réseau via des ports ; il a une IP (adresse internet)
- un **serveur logiciel** est un processus exécuté en tâche de fond d'un serveur machine qui communique avec des **clients (logiciels)** via un port particulier.
- Une **machine virtuelle** est un programme qui simule, sur une machine hôte, un autre ordinateur.
- Un **client** (logiciel) est un programme qui communique avec un serveur (logiciel)

Exemple

Un serveur web est un processus (Apache par exemple) qui communique sur le port 80 d'un serveur machine. Si ce serveur machine a pour IP 163.12.9.10, alors tout client web peut s'adresser au serveur web à l'adresse 163.12.9.10 :80.

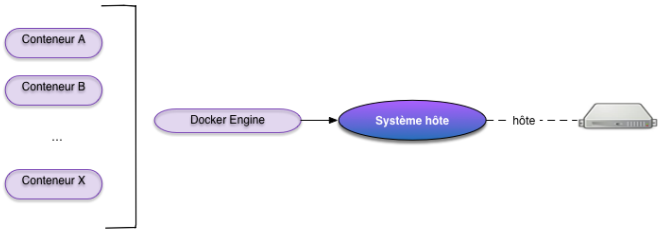
Illustration

Voici un exemple de système distribué.



Docker

Docker permet de créer des **conteneurs**, des machines virtuelles extrêmement légères.



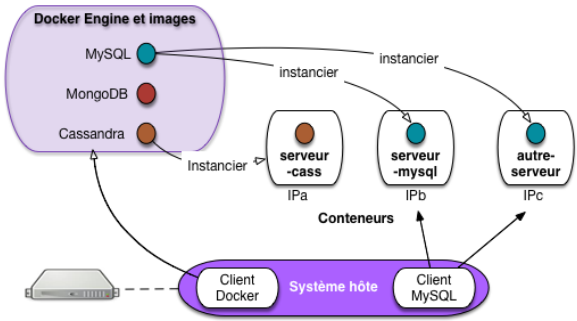
Existe sous Linux, ainsi que MacOS et Windows.

Vocabulaire

- Le **système hôte** est le système d'exploitation principal gérant votre machine ;
- **Machine Docker** (ou **Docker VM**), désigne le système Linux où Docker s'exécute ;
- **Docker engine** est le programme qui gère les conteneurs ; il s'exécute dans la machine Docker.
- Un **conteneur** est une partie autonome de la **Docker VM**, se comportant comme une machine indépendante.
- On gère les conteneurs soit avec la ligne de commande (**Docker CLI**), soit une interface comme le **Docker desktop** (plus facile)

Les images

Image = système pré-configuré, prêt à l'emploi pour un service particulier (par exemple un serveur MySQL, ou MongoDB).



Une image s'installe très facilement. C'est une sorte de moule pour créer ("instancier") des conteneurs.

Comment communiquer avec un conteneur ?

J'ai lancé un conteneur MySQL. Mon serveur tourne, sur le port 3306, **dans le conteneur**.

Question : comment mon client MySQL peut-il communiquer avec ce serveur ?

Réponse : *il faut associer un port de la machine Docker avec le port 3306 du conteneur* (*port forwarding*).

Par exemple,

- je vais associer le port 6603 de la machine Docker avec le port 3306 du conteneur `serveur-mysql`
- je vais associer le port 6604 de la machine Docker avec le port 3306 du conteneur `autre-serveur`

À l'action !

Et voilà pour les principes.

Pour la suite : mise en pratique avec installation de systèmes NoSQL.

- CouchDB
- MongoDB
- Cassandra
- ElasticSearch

Votre mission : effectuer les manipulations Docker décrites dans le support de cours, jusqu'à ce que vous soyez à l'aise **en comprenant bien ce qui se passe.**