DEVOPS I : TOUT SAVOIR DE DOCKER EN QUELQUES HEURES

Création et premières utilisations des conteneurs

JORDAN ASSOULINE

OBJECTIFS DE LA SECTION

Objectifs

- Comprendre les commandes de base (docker version, docker info)
- Lancer un serveur web (docker run)
- Lister les conteneurs (docker ls)
- Arrêter et supprimer un conteneur (docker stop, docker rm)
- Afficher les logs et les processus d'un conteneur (docker logs, docker top)
- Que se passe-t-il lors d'un "docker container run" ?
- Qu'est-ce que la virtualisation ?
- Les conteneurs c'est quoi en fait ?
- Exercice : Création de plusieurs conteneurs
- Obtenir plus d'informations sur nos conteneurs
- Naviguer à l'intérieur de son conteneur

JORDAN ASSOULINE

LES COMMANDES DE BASE

JORDAN ASSOULINE

La commande « docker version »

- **`docker version`**
  - Cette commande nous permet d'afficher la version du client et du serveur (aussi appelé Engine) de Docker.
  - Elle permet notamment de vérifier le bon fonctionnement de Docker

- https://docs.docker.com/engine/reference/commandline/version/

La commande « docker info »

- **`docker info`**
  - Cette commande nous permet d'afficher plus de détails concernant le serveur Docker (aussi appelé le docker Engine).
    - Le nombre de conteneurs actuellement en fonctionnement
    - Le nombre d'images stockées dans Docker

- https://docs.docker.com/engine/reference/commandline/info/

La commande « docker »

- **`docker`**
  - Cette commande permet d'afficher l'aide et nous montrer l'ensemble des commandes qu'il est possible de faire.
  - Elle affiche notamment les « management commands » permettant une meilleure lisibilité (disponibles depuis 2017).
  - Pour que les anciens utilisateurs ne soient pas perdus, on peut utiliser les commandes avec les « management commands » ou sans les utiliser:
    - docker container run
    - docker run

- https://docs.docker.com/engine/reference/commandline/docker/

PREMIÈRES MANIPULATIONS SUR LES CONTENEURS

Lancement d’un serveur web

JORDAN ASSOULINE

Quelle différence entre « image » et « conteneur » ?

- Une **image** est constituée des fichiers binaires, des librairies et du code source qui compose l'application.

- Un **container** correspond à l'instance, en cours d'exécution, de l'image.

- Docker Hub (https://hub.docker.com) correspond au registre des images par défaut (un peu comme github pour le code source)

Lancer un conteneur type « serveur web » avec l'option publish

- **`docker container run --publish 80:80 nginx`**
  - Télécharge l'image "**nginx**" depuis le docker hub dans sa version la plus récente (c'est comme si on avait noté nginx:latest)
  - Démarre un nouveau conteneur à partir de cette image
  - La partie "**publish 80:80**" expose le port local 80 de la machine hôte, et redirige tout le trafic à l'application exécutée dans le conteneur sur son port 80.
    - Ouvre le port 80 sur votre machine
    - Redirige le trafic à destination du port 80 de votre machine vers le port 80 du conteneur
  - Vous pouvez obtenir une "bind error" si le port hôte (celui de gauche) est déjà utilisé par quelque chose y compris un autre conteneur.
  - https://docs.docker.com/engine/reference/commandline/run/

Lancer un conteneur type « serveur web » avec l'option detach

- **`docker container run --publish 80:80 --detach nginx`**
  - L'option detach permet d'exécuter le démarrage du container en arrière-plan, vous donnant la main pour effectuer d'autres actions.
  - https://docs.docker.com/engine/reference/commandline/run/

PREMIÈRES MANIPULATIONS SUR LES CONTENEURS

Lister les conteneurs

JORDAN ASSOULINE

Lister les conteneurs sur notre serveur

- **`docker container ls`**
  - Liste les conteneurs qui sont actuellement en cours d'exécution
- **`docker container ls -a`**
  - Liste tous les conteneurs, y compris ceux qui ont été arrêtés
- Remarque : on peut aussi utiliser la commande **`docker ps`** directement pour obtenir le même résultat (ancienne syntaxe)

Quelques informations supplémentaires

- A chaque fois que vous lancez un conteneur en utilisant la commande `docker container run`, vous créer un nouveau conteneur avec un nom généré aléatoirement (sous la colonne NAMES de la commande `docker container ls –a`) sauf si on lui précise nous-même le nom à utiliser.

Lancer un conteneur type « serveur web » avec l'option name

- **`docker container run --publish 80:80 --detach --name nginx_webserver nginx`**
  - L'option name permet de spécifier un nom et de l'associer à ce conteneur, ce qui est beaucoup plus simple par la suite pour pouvoir interagir avec lui et lui envoyer des commandes (plutôt que d'utiliser son id).
  - https://docs.docker.com/engine/reference/commandline/run/

PREMIÈRES MANIPULATIONS SUR LES CONTENEURS

Arrêter et supprimer un conteneur

JORDAN ASSOULINE

Arrêter un conteneur :

- **`docker container stop <NAME/ID>`**
  - Permet d'arrêter un container (il ne sera donc visible que via la commande `docker container ls -a`)

Supprimer des conteneurs

- **`docker container rm <NAME> <NAME2> ...`**
  - Permet de supprimer définitivement un container (ou une liste de containers)
  - Remarque : seuls les containers déjà stoppés peuvent être supprimés

- **`docker container rm -f <NAME> <NAME2> ...`**
  - Permet de supprimer définitivement un container même si celui-ci n'a pas été stoppé au préalable grâce à l'option -f

PREMIÈRES MANIPULATIONS SUR LES CONTENEURS

Afficher les logs et les processus d'un conteneur

JORDAN ASSOULINE

Afficher les logs d'un conteneur

- **`docker container logs <NAME/ID>`**
  - L'option logs permet d'afficher les logs du conteneur indiqué soit par son ID, soit par son nom.
  - https://docs.docker.com/engine/reference/commandline/logs/

Afficher le processus exécuté sur un conteneur

- **`docker container top <NAME/ID>`**
  - L'option top permet d'afficher le processus qui est actuellement en cours d'exécution sur le conteneur.
  - https://docs.docker.com/engine/reference/commandline/top/

QUE FAIT RÉELLEMENT DOCKER ?

JORDAN ASSOULINE

Que se passe-t-il réellement lors d'un 'docker container run' ?

- Docker va d'abord chercher l'image que l'on souhaite utiliser dans son cache local
- S'il ne la trouve pas, il va faire une recherche sur le Docker Hub par défaut (dépôt que l'on peut modifier)
- Télécharge la dernière version de l'image (sauf si on lui précise la version à télécharger) et la stocke dans son cache
- Création d'un nouveau conteneur basé sur l'image spécifiée et se prépare à l'exécuter

Que se passe-t-il réellement lors d'un 'docker container run' ? (2)

- Attribution d'une adresse IP virtuelle sur un réseau privé à l'intérieur du serveur Docker (docker engine)

- Si l'on utilise l'option --publish 80:80, il va ouvrir le port 80 de votre ordinateur/serveur, puis rediriger le trafic reçu sur ce port à destination du port 80 du conteneur

- (Démarre le conteneur en utilisant la « CMD » spécifiée dans le fichier Dockerfile)

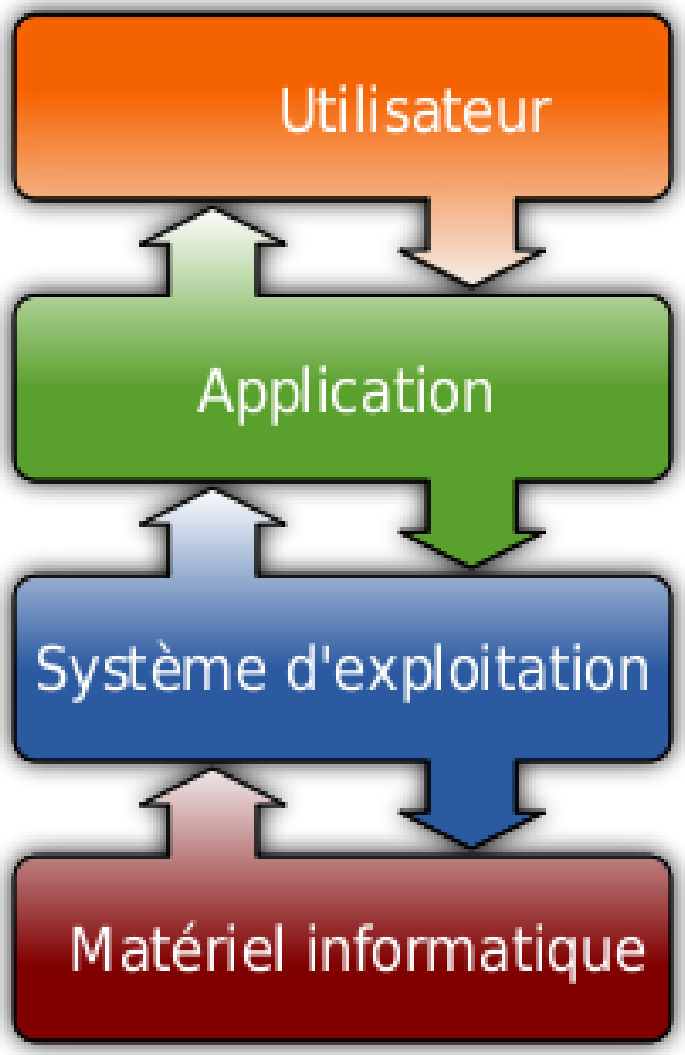
QU'EST-CE QUE LA VIRTUALISATION ?

JORDAN ASSOULINE

Définition de la virtualisation

- “La virtualisation est un mécanisme informatique qui consiste à **faire fonctionner plusieurs systèmes**, serveurs ou applications, **sur un même serveur physique**.”

JORDAN ASSOULINE

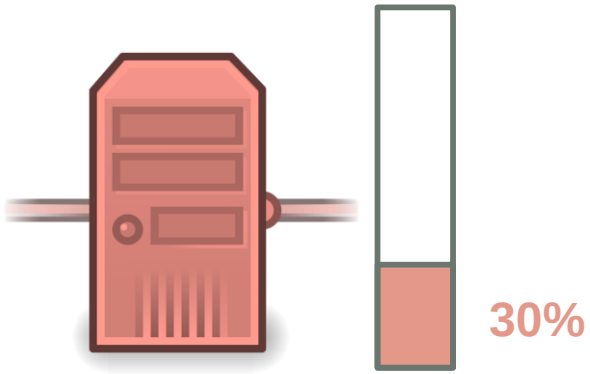


Définition d'un système d'exploitation

- Un OS est un logiciel qui pilote les dispositifs matériels et reçoit des instructions de l'utilisateur ou d'autres logiciels.

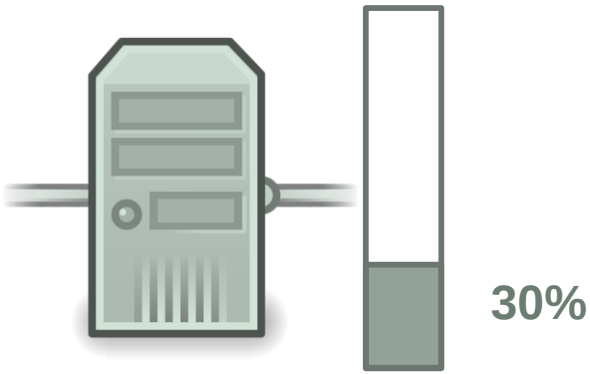
Avant la virtualisation

Chaque service était hébergé sur un serveur différent, n'utilisant qu'une fraction du processeur et de la RAM de chaque machine…



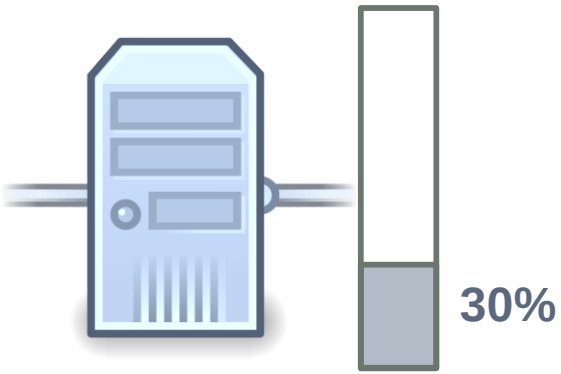
30%

Serveur Web



30%

Serveur Messagerie



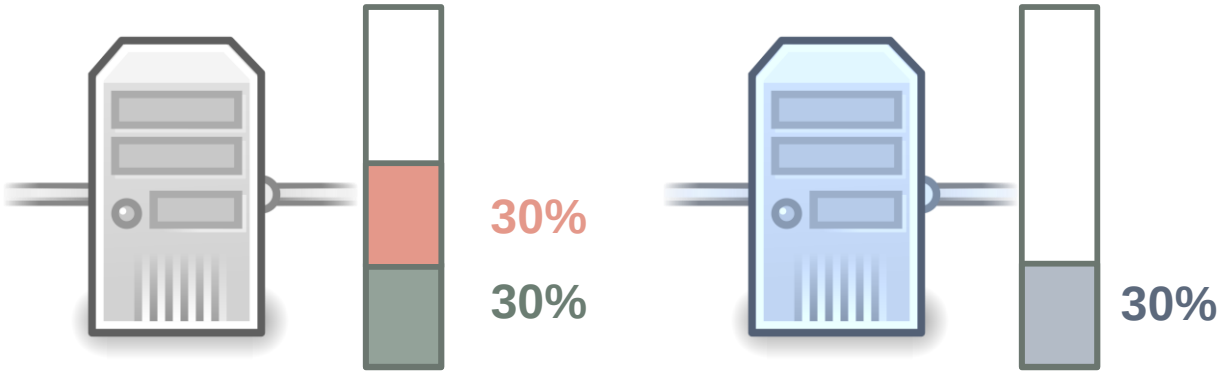
30%

Annuaire (AD/LDAP)

JORDAN ASSOULINE

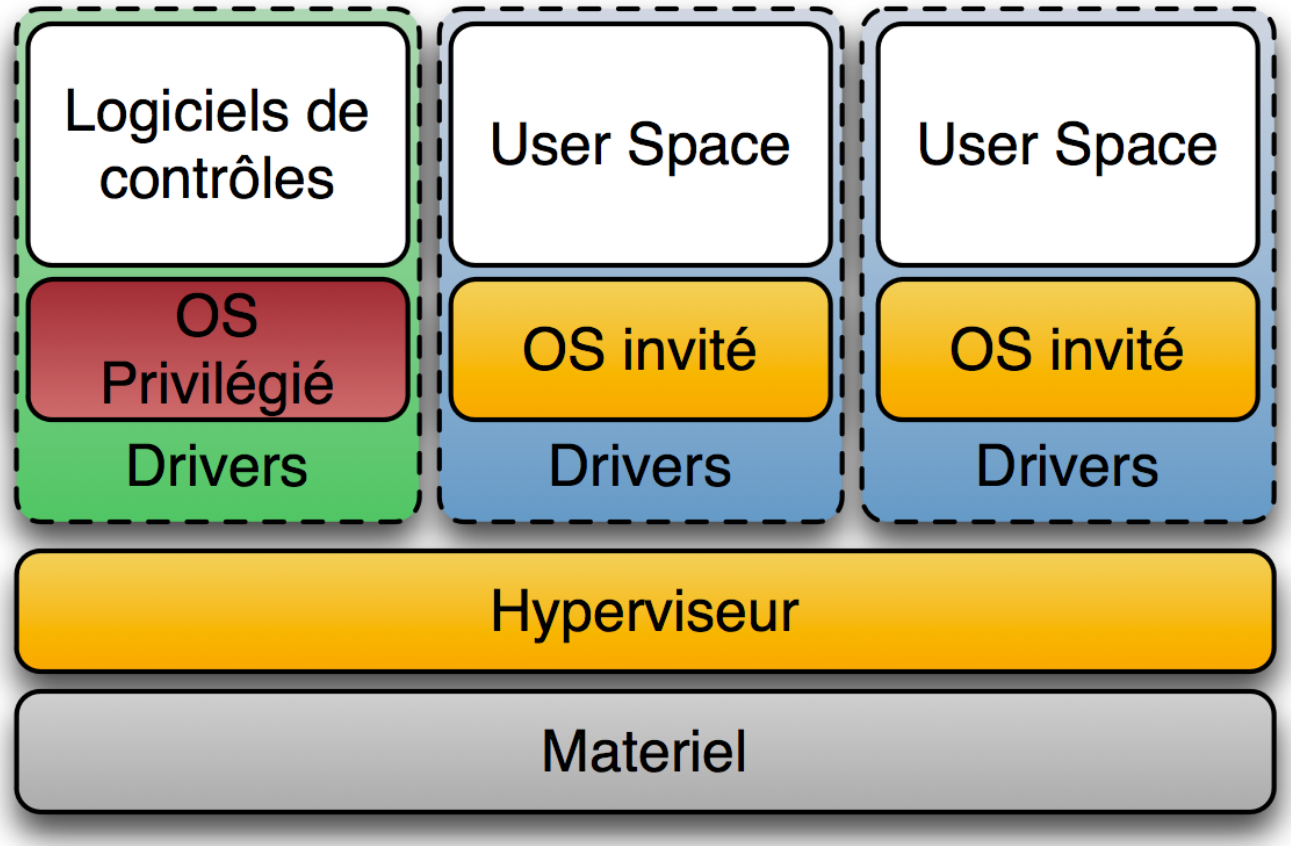
Avec la virtualisation

Plusieurs services peuvent être hébergés sur la même machine, afin d'utiliser au maximum les ressources disponibles



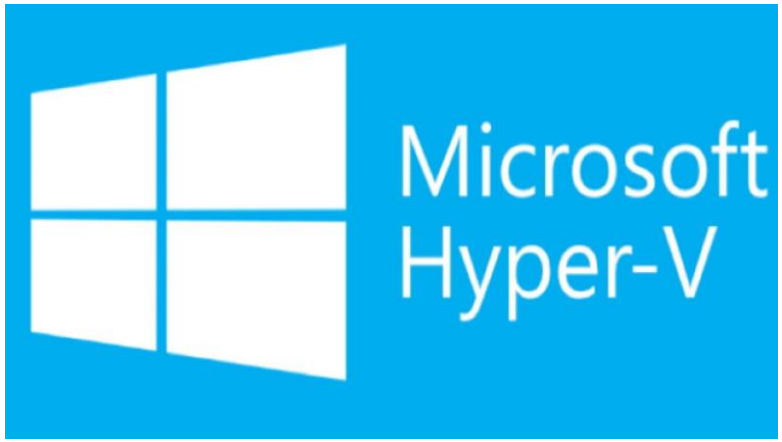
JORDAN ASSOULINE

Hyperviseur de type 1



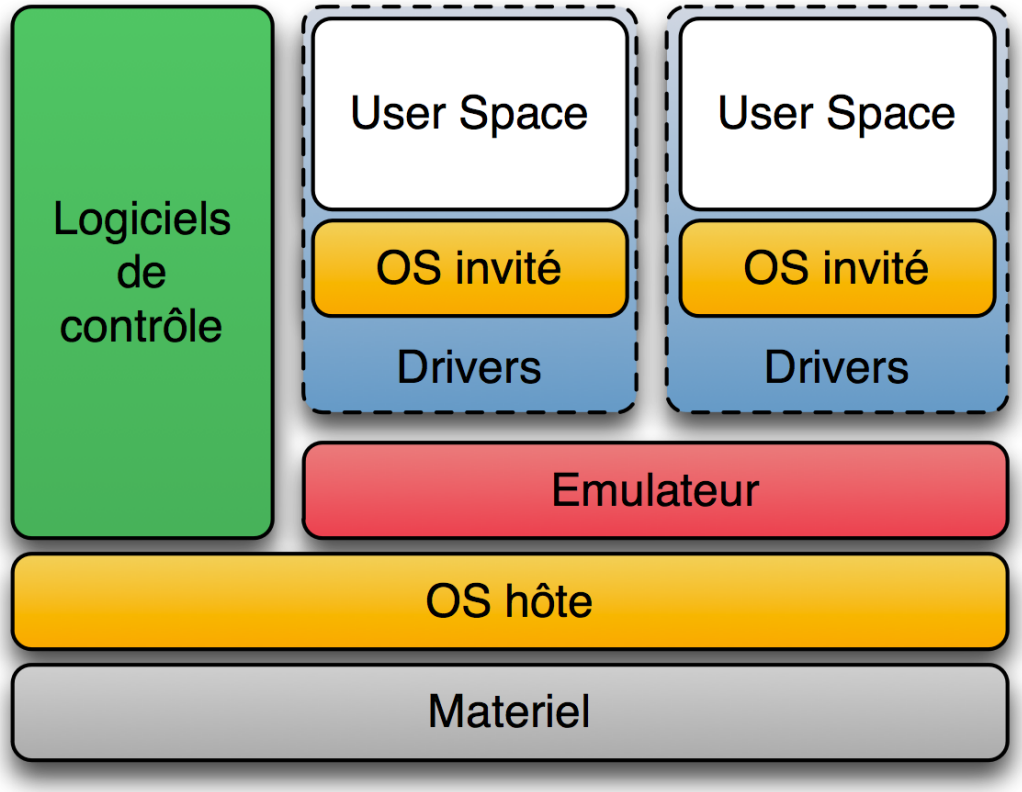
JORDAN ASSOULINE

Hyperviseur de type 1



JORDAN ASSOULINE

Hyperviseur de niveau 2



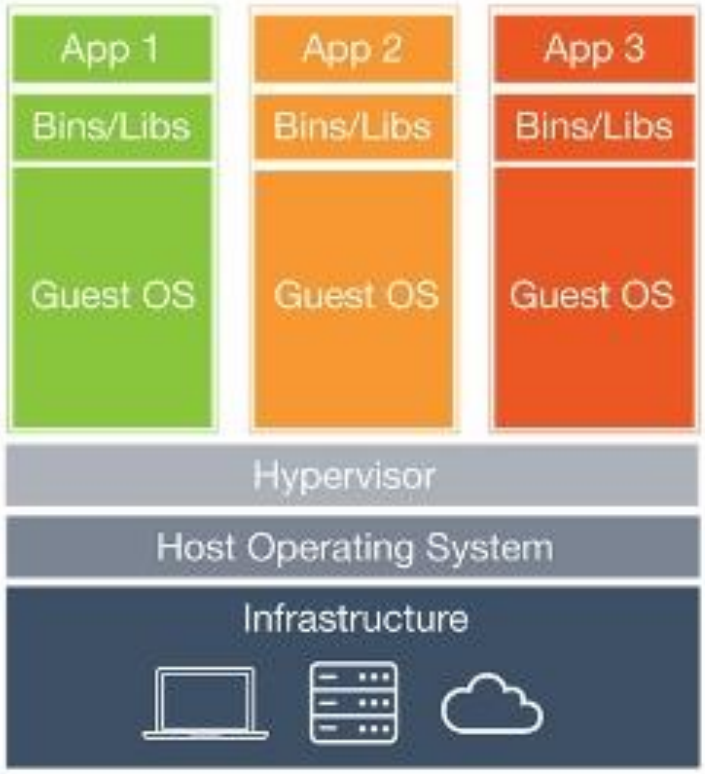
JORDAN ASSOULINE

Hyperviseur de type 2

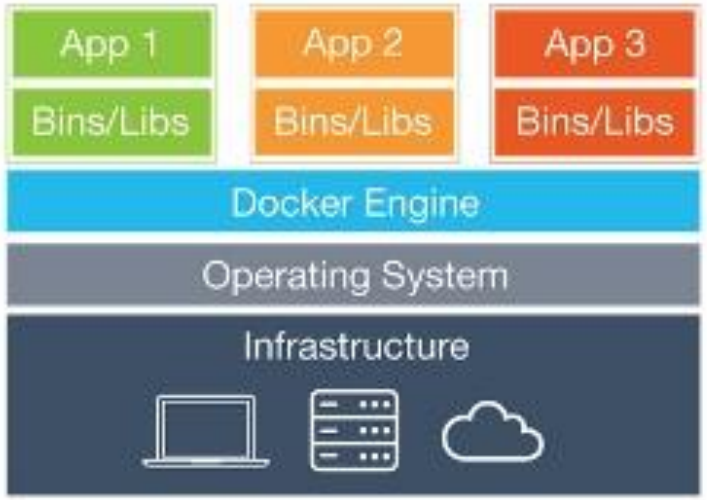
JORDAN ASSOULINE

LES CONTENEURS C'EST QUOI ?

Une différence subtile mais essentielle !



Virtual Machines



Containers

JORDAN ASSOULINE

Qu’est-ce qu’une image ?

- Une image correspond aux fichiers binaires et aux dépendances, ainsi qu'aux métadonnées concernant la façon dont il faudra l'exécuter.

- Définition officielle : "**Une image est une collection ordonnées de modifications du filesystem root et les paramètres d'exécution correspondants pour l'exécuter à l'intérieur d’un conteneur.** "

- Il n'y a pas d'OS complet, pas de kernel et pas de modules kernel (drivers par exemple).

Les conteneurs ne sont pas des VMs !

- Les conteneurs ne sont que de simples processus (ou démons)
- Ils sont légers, portables et permettent notamment à des applications d’être packagées et déplacées rapidement.
- L'image peut être aussi petite qu'un simple fichier (par exemple lorsqu'on a une app en Go), mais aussi assez grande lorsqu'il s'agit d'une distribution Ubuntu avec APT, apache et PHP installés.

EXERCICE 1

Instructions

Utiliser plusieurs conteneurs en même temps

- Lancez trois conteneurs : **nginx**, **mysql** et **httpd**

- Lancez les trois conteneurs en utilisant les deux options suivantes à chaque fois :
  - **--detach**
  - **--name**

- Au niveau des ports d'écoute, vous devrez configurer les conteneurs de la manière suivante :
  - nginx en **80:80**
  - httpd en **8080:80**
  - mysql en **3306:3306**

Utiliser plusieurs conteneurs en même temps

- Lorsque vous lancerez le conteneur mysql, utilisez l'option **`--env`** pour lui faire passer la variable : **`MYSQL_RANDOM_ROOT_PASSWORD=yes`**
  - https://hub.docker.com/_/mysql

- Utiliser la commande **`docker container logs`** sur mysql pour récupérer le mot de passe root qui a été généré aléatoirement.

- Terminez en utilisant les commandes **`docker container stop`** et **`docker container rm`** pour tout nettoyer.

EXERCICE 1

Correction

Les commandes à utiliser

- Pour nginx :
  - `docker container run --detach --publish 80:80 --name nginx_server nginx`
- Pour apache :
  - `docker container run --detach --publish 8080:80 --name apache_server httpd`
- Pour mysql :
  - `docker container run --detach --publish 3306:3306 --name mysql_server --env MYSQL_RANDOM_ROOT_PASSWORD=yes mysql`
  - `docker container logs mysql_server`

JORDAN ASSOULINE

OBTENIR PLUS D'INFORMATIONS SUR NOS CONTENEURS

JORDAN ASSOULINE

Afficher le processus exécuté sur un conteneur

- **`docker container top <NAME/ID>`**
  - L'option top permet d'afficher le processus qui est actuellement en cours d'exécution sur le conteneur.
  - https://docs.docker.com/engine/reference/commandline/top/

Utiliser la commande « inspect»

- **`docker container inspect <NAME>`**
  - Donne toutes les informations au format json à propos du container (son adresse IP, les interfaces configurées, etc...)
  - https://docs.docker.com/engine/reference/commandline/inspect/

Utiliser la commande « stats »

- **`docker container stats <NAME>`**
  - Affiche les statistiques concernant l'utilisation de la RAM, du CPU, etc...
  - https://docs.docker.com/engine/reference/commandline/stats/

ALLER À L'INTÉRIEUR DE SON CONTENEUR

Utiliser l'option « it » pour interagir avec son conteneur

- **`docker container run -it --name webserver --publish 80:80 nginx bash`**
  - L'option **t** permet de simuler un vrai terminal, exactement comme le ferait SSH
  - L'option **i** permet de garder la session ouverte pour recevoir l'output des commandes que l'on tape
  - La commande **bash** donne un terminal existant à l'intérieur du container afin de pouvoir y exécuter des commandes interprétables

  - https://docs.docker.com/engine/reference/commandline/run/

Utiliser une commande différente de celle par défaut

```
λ docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
700b8b686a0f        nginx               "nginx -g 'daemon of…"   12 seconds ago      Exited (0) 6 seconds ago                        webserver2
3fa672c1c456        nginx               "bash"                   28 seconds ago      Exited (127) 23 seconds ago                     webserver
```

- Si on lance le conteneur nginx sans spécifier de CMD, il va utiliser celle par défaut (« nginx –g »)
- Si on précise la commande « bash », il va utiliser celle-ci à la place, et dés qu'on sortira du terminal bash, il arrêtera le conteneur

JORDAN ASSOULINE

Exécuter une commande à l'intérieur de son conteneur avec « docker exec »

- **`docker container exec  -it -<NAME> bash`**
  - Exécute la commande bash en mode interactif à l'intérieur du container
  - Attention, par exemple sur l'image alpine, bash n'est pas disponible, et il faut plutôt utiliser **sh**

  - https://docs.docker.com/engine/reference/commandline/exec/