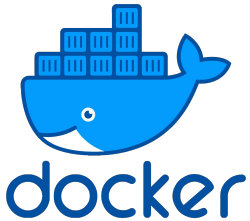


Formation kubernetes

25 - 27 Septembre 2023.

Tour de table des stagiaires.

- Nom & prénom
- Expériences
- Précisez vos attentes pour cette formation

Présentation du formateur - Didier Curvier MALEMBE.

- **Ingénieur en Informatique | DEVOPS CERTIFIED**
  - > Capgemini Technology Services
    - > SNCF Connect - SNCF RM - DB etc …
- **Consultant Freelance**
  - > Secteur public / TPE-PME
- **Formateur partenaire**
  - > ORSYS Formation
  - > FITEC
  - > AFPA
  - > PMN > **IB Cegos**
- **Co-founder ELITIS CONSULTING**

Les objectifs de la formation.

3 objectifs principaux :

- Initiation à Docker et gestion avancée des conteneurs.
- Initiation à l'exploitation de kubernetes.
- La gestion des paquets avec Helm.

La virtualisation

Qu’est ce que la virtualisation ?

De façon générale, la **virtualisation** est le processus qui consiste à créer une version logicielle ou virtuelle d’une entité physique telles que des applications ou des serveurs.

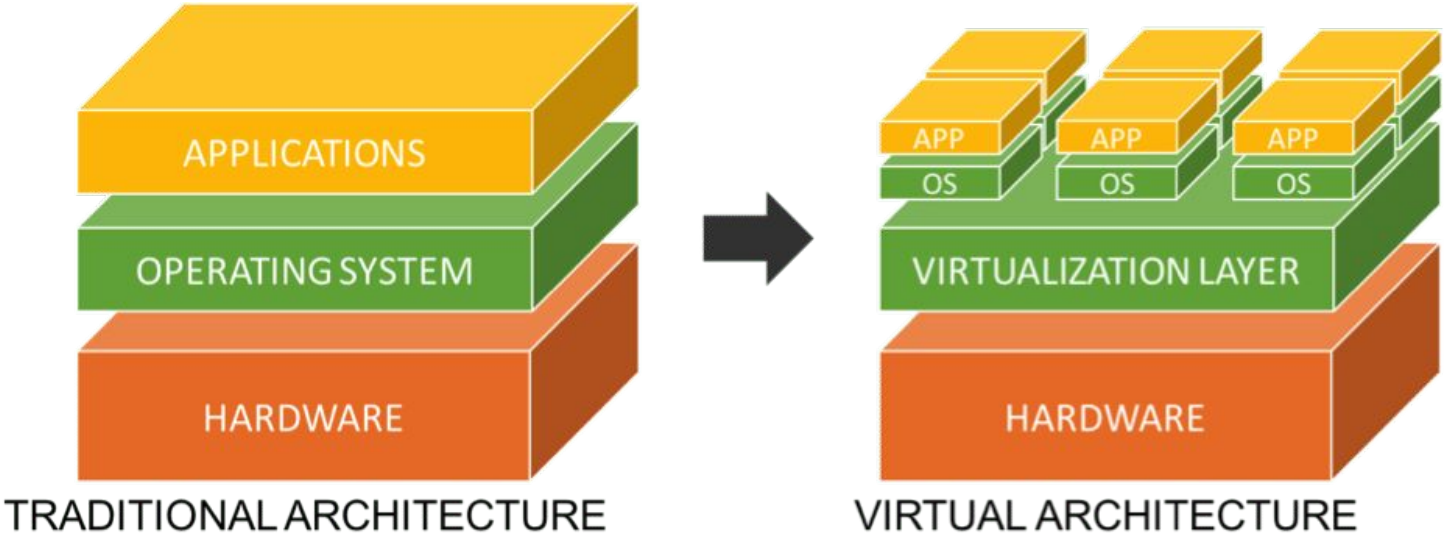
Au lieu d’avoir un serveur sur lequel tournent plusieurs applications, on crée plusieurs serveurs mutualisés à partir d’un serveur physique. On utilise pour cela un **hyperviseur**.

Les VMs - Machines virtuelles.

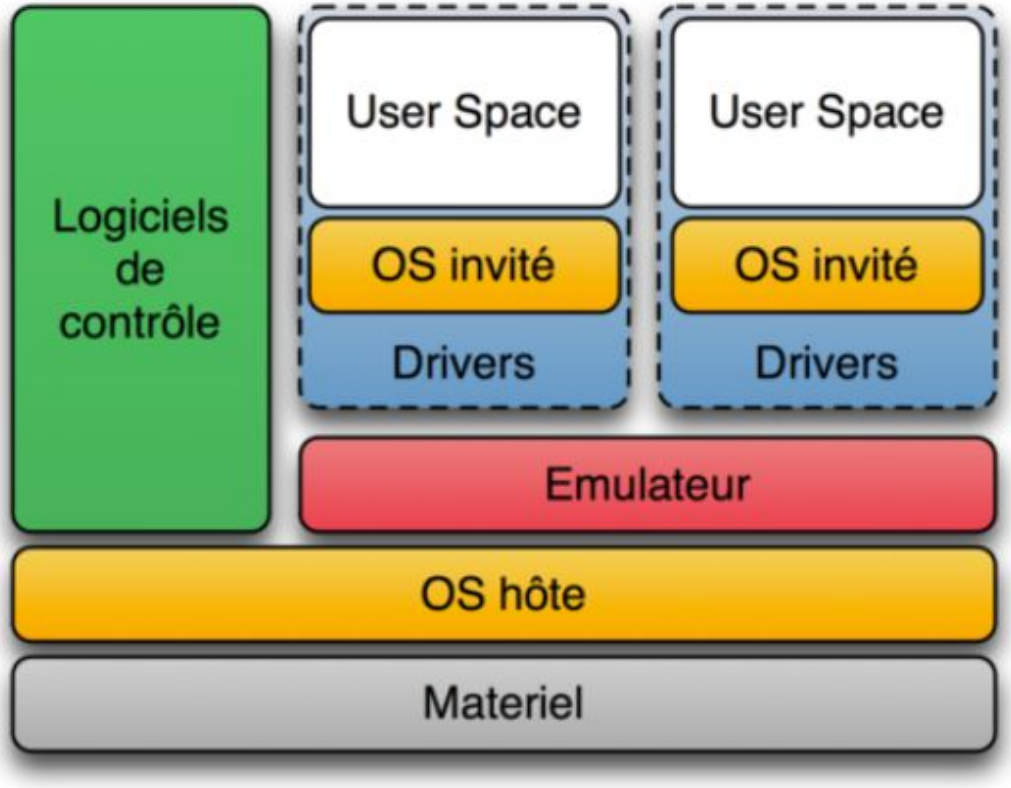
- Les machines issues de la virtualisation sont dites ***“machines virtuelles”***.
- Les machines virtuelles ont leurs propres systèmes d’exploitation.

- Elles disposent aussi de ressources matérielles évolutives selon les besoins (les capacités allouées aux ressources sont modifiables).

Virtualisation



Virtualisation

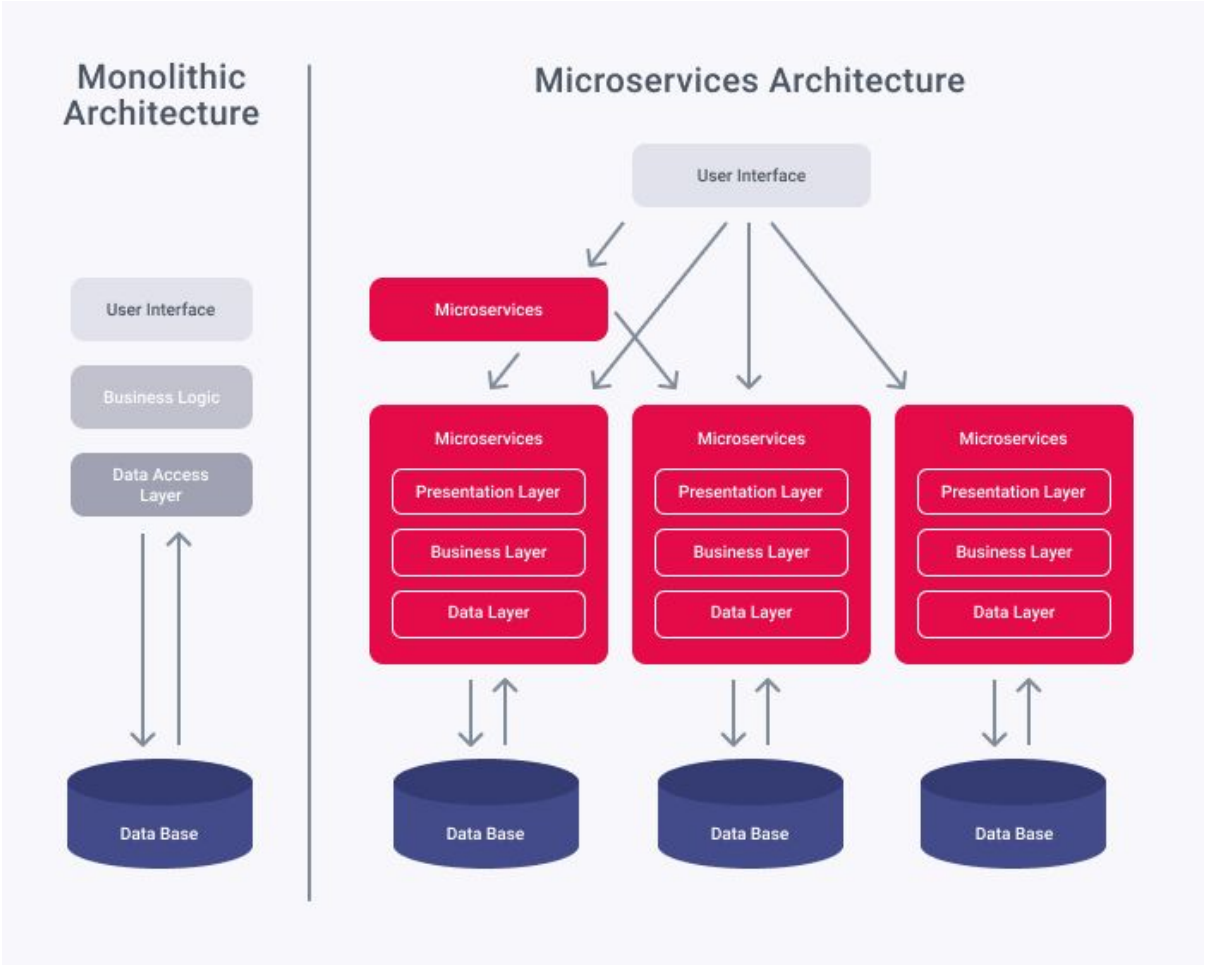


Avantages & inconvénients.

- Manipulations simplifiées.
- Réduction du nombre de serveurs physiques
- Réduction des dépenses liées à l'infrastructure
- Facilitation du Plan Reprise d'Activité

- La couche d'abstraction matérielle à tendance à réduire les performances.
- L'OS consomme énormément de ressources, au détriment des applications qui s'exécutent dessus.

De la virtualisation à la conteneurisation.


Monolithic Architecture
User Interface
Business Logic
Data Access Layer
Data Base
Microservices Architecture
User Interface
Microservices
Microservices
Presentation Layer
Business Layer
Data Layer
Microservices
Presentation Layer
Business Layer
Data Layer
Microservices
Presentation Layer
Business Layer
Data Layer
Data Base
Data Base
Data Base

Conteneurisation KESAKO ?

- La conteneurisation consiste à rassembler le code du logiciel et tous ses composants (bibliothèques, frameworks et autres dépendances) de manière à les isoler dans leur propre « conteneur ».

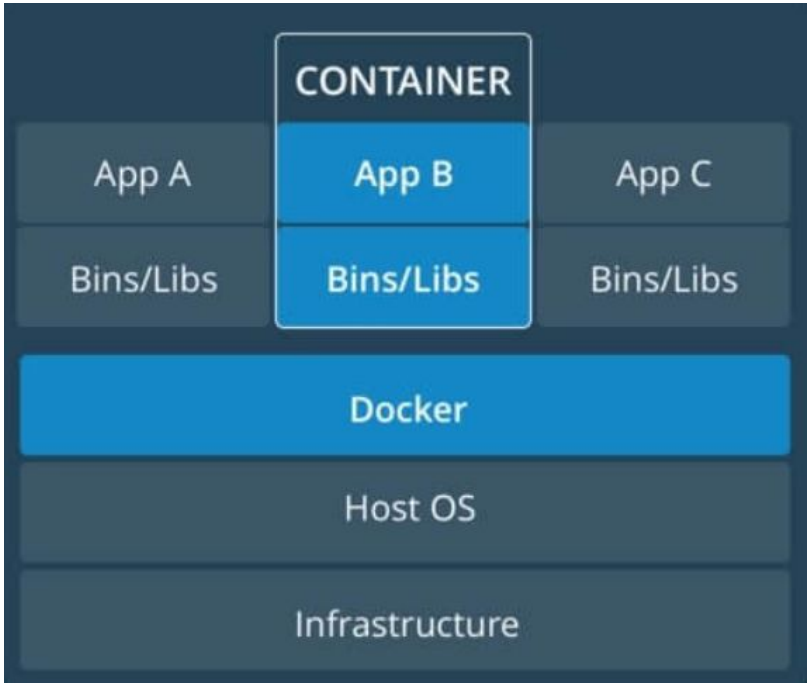
- La conteneurisation est l'utilisation des conteneurs Linux pour déployer des applications.

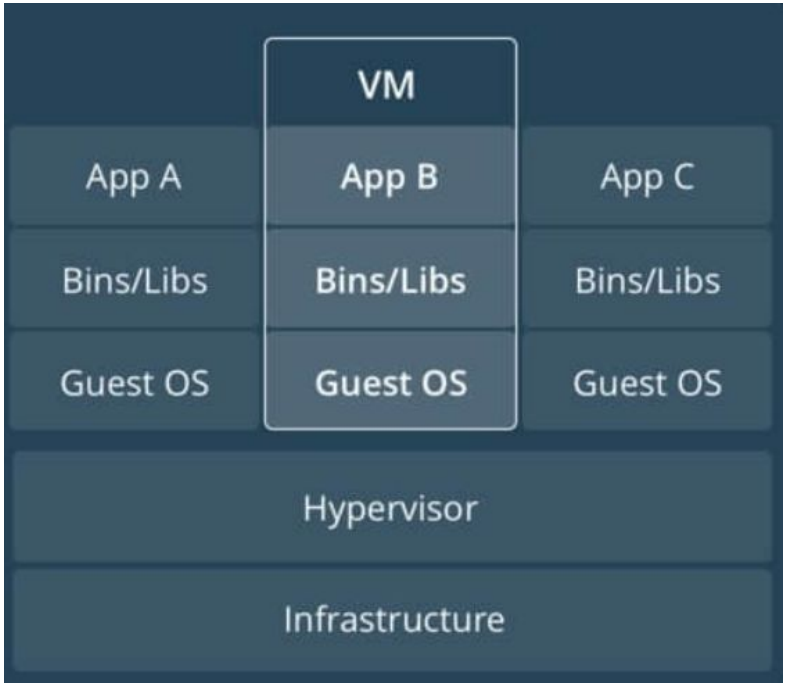
Conteneurisation VS virtualisation

S’agissant de la virtualisation, les machines virtuelles accèdent aux ressources de l’ordinateur hôte via un **hyperviseur**. On parle d’isolation matérielle *(CPU/RAM/Disque)*.

En revanche dans le cas de la **conteneurisation**, le conteneur s'exécute sous linux et va partager le noyau de la machine hôte avec d’autres conteneurs. On parle aussi d’isolation du système d’exploitation.

Conteneurisation VS virtualisation.

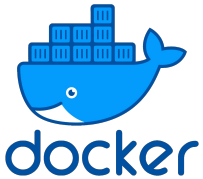




Pourquoi conteneuriser ?

1. Les machines virtuelles intègrent un système d'exploitation qui peut peser lourd.
2. Les conteneurs réalisent des appels au système d'exploitation pour exécuter ses applications.
3. Les conteneurs nécessitent moins de ressources.
4. Il est beaucoup plus facile de déplacer des conteneurs dans différents environnements.

Introduction à la technologie Docker.



Docker KESAKO ?

- Docker est une plateforme fondée par **Solomon Hykes** et officiellement lancée en 2013 pour faciliter la **conteneurisation des applications**.

- Docker permet d'embarquer une application avec l'ensemble de ses **dépendances** dans un processus isolé qu'on appelle **conteneur**. Ce conteneur peut ensuite être exécuté sur n'importe quelle environnement (machine/système d'exploitation) compatible avec le docker.

Avantages de la conteneurisation.

- Plus facile à manipuler
- Meilleure gestion des ressources
- Répondre pleinement au principe de l'architecture des microservices

Docker est connue comme la solution de conteneur la plus populaire, dispose d'une excellente documentation (et est bien suivie par la communauté) et constitue un choix sûr utilisé par de nombreuses entreprises.

- Démarrer/Arrêter les conteneurs
- Télécharger/Télécharger des images dans un registre
- Assez simple pour un environnement de développement

Installation de Docker.

Installation de Docker.

Il existe deux éditions de Docker :

- **Docker Entreprise Edition (CE)**
- **Docker Community Edition (EE)**

Dans le cadre de cette formation nous utiliserons la Community Edition car c’est la version gratuite idéale pour l’apprentissage.

Travailler avec les images docker.

Qu’est ce qu’une image Docker ?

Une image docker est un package qui inclut tout ce qui est nécessaire à l'exécution d’une application, à savoir :

- Le code
- L'exécution
- Les variables d’environnement
- Les fichiers de configuration
- etc

Qu’est ce qu’une image Docker ?

- Une image docker est créée à partir d’un Dockerfile.
- Une image docker comprend plusieurs couches.
- Les différentes couches comprennent l’application et ses dépendances (binaires - bibliothèques…).

- **Un conteneur est une image en cours d'exécution.**

Commandes usuelles.

Lister les commandes possibles - docker help

docker help : Liste des commandes possibles.

docker info : Afficher les détails sur l'installation.

docker volume : Cette commande, nous fournit plusieurs informations concernant les spécifications du moteur Docker.

docker --version : Consulter la version docker.

Lister les images docker - docker image ls

`docker images` ou `docker image ls` : permet de répertorier les images docker téléchargées.

```
C:\Users\Lenovo>docker images
REPOSITORY                                    TAG             IMAGE ID       CREATED         SIZE
spring-boot-docker                            spring-docker   412a030f013a   3 months ago    671MB
docker.elastic.co/elasticsearch/elasticsearch 8.6.2           04485c81cc2d   4 months ago    1.29GB
gcr.io/k8s-minikube/kicbase                   v0.0.37         01c0ce65fff7   5 months ago    1.15GB
docker/getting-started                        latest          3e4394f6b72f   6 months ago    47MB
```

La commande renvoie les informations suivantes :

Lister les images docker - docker image ls

- **REPOSITORY :** Correspond au nom de l'image.
- **TAG :** Permet d'afficher la version de l'image.
- **IMAGE ID :** L'identifiant unique de chaque image.
- **CREATED :** Date de la dernière modification de l'image.
- **SIZE :** Taille de l'image .

Supprimer une image Docker - docker rmi -f image_id.

Suppression avec l'image ID:

```
C:\Users\Lenovo>docker rmi -f  412a030f013a
Untagged: spring-boot-docker:spring-docker
Deleted: sha256:412a030f013adbdfb34b3b761b6e3e99db94aaa775e68305b09a91d5accc284b
```

Docker Hub Registry

Qu’est ce que le Docker Hub Registry ?

Le **Docker Hub Registry** est un service de registre d'images Docker public et centralisé fourni par Docker.

C'est l'endroit où les utilisateurs peuvent **partager, stocker et récupérer des images Docker**.

Le **Docker Hub Registry** permet aux développeurs de télécharger des images préexistantes créées par d'autres utilisateurs, ce qui facilite grandement le processus de développement et de déploiement d'applications basées sur Docker.

https://hub.docker.com/

Access the world's largest library of container images

busybox
Official
1B+

ubuntu
Official
1B+

python
Official
1B+

postgres
Official
1B+

httpd
Official
1B+

memcached
Official
1B+

mysql
Official
1B+

traefik
Official
1B+

mariadb
Official
1B+

docker
Official
1B+

See all Docker Official Images

Lister les images sur Docker Hub Registry.

Pour vérifier la disponibilité d'une image sur Docker Hub Registry on peut rechercher son nom dans la barre de recherche.

La deuxième manière pour lister les images disponibles dans le Docker hub Registry,c'est de passer par la ligne de commande :

docker search ubuntu ou encore :

docker search --filter "is-official=true" ubuntu

Récupérer une image depuis le DHR - docker pull

On utilise la commande docker pull

ie : **`docker pull ubuntu`**

Travailler avec les conteneurs docker.

Qu’est ce qu'un conteneur docker ?

Un **conteneur** est une image en cours d’exécution. En d’autres termes, un conteneur c’est ce que devient une image lorsqu'elle est exécutée.

Dans ces **conteneurs** on peut retrouver généralement un ou plusieurs programme(s) avec toute leurs dépendances de manière à les maintenir isolées du système hôte sur lequel elles s'exécutent.

Commandes usuelles.

Créer un conteneur - docker run.

On utilise la commande suivante :

`docker run [OPTIONS] <IMAGE_NAME ou ID>`

La commande docker run possède de nombreuses options consultables.

`-t` : Allouer un pseudo TTY

`-i` : Garder un STDIN ouvert.

`-d` : Exécution en arrière-plan du conteneur et affichage de l'ID du conteneur.

Lister les conteneurs - docker container ls.

`docker container ls` ou **`docker ps`**

Cette commande renvoie les informations suivantes :

CONTAINER ID : ID du conteneur

IMAGE : L'image sur laquelle est basée le conteneur.

COMMAND : Dernière commande lancée lors de l'exécution de votre image.

CREATED : date de création du conteneur.

STATUS : Statut de votre conteneur, voici une

Supprimer un conteneur - docker rm .

```
docker rm <CONTAINER   NAME ou ID>
```

```
docker rm mynginx
```

Images Docker avec le Dockerfile

Dockerfile

Généralement pour construire une image on se place directement dans le dossier avec le Dockerfile et les éléments de contexte nécessaire (programme, config, etc), le contexte est donc le caractère . il est obligatoire de préciser un contexte.

Exemple : docker build -t mondebian .

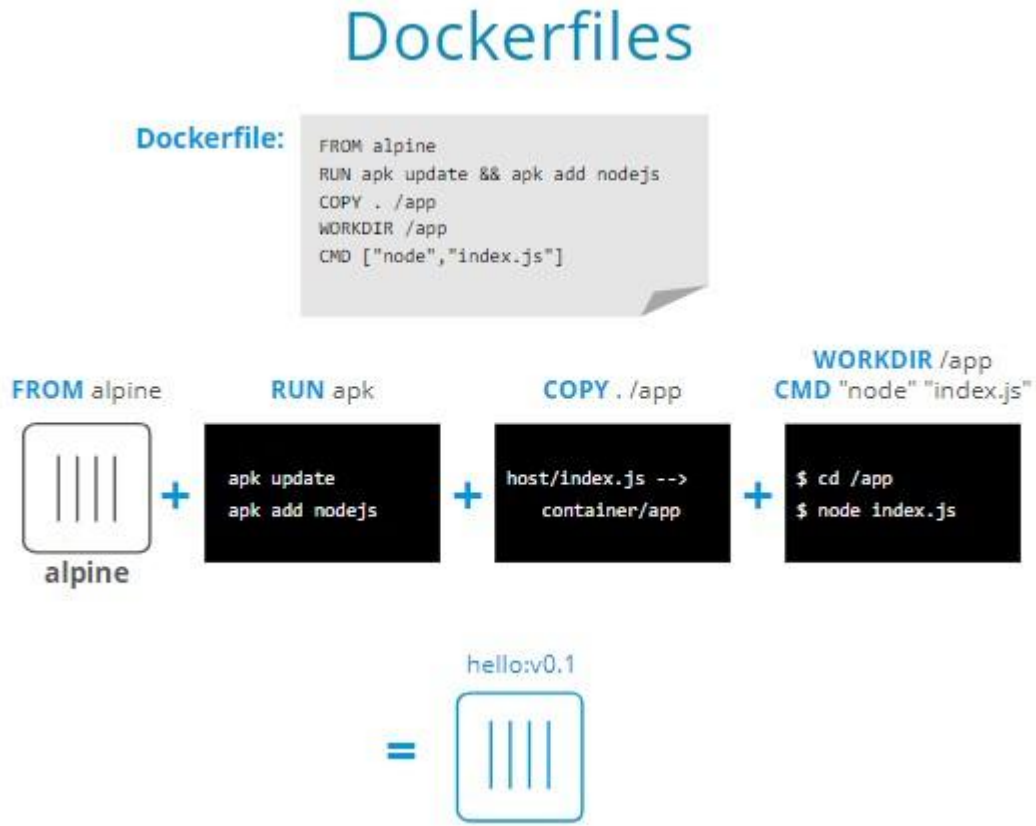
Exemple de Dockerfile

```
FROM debian:latest

RUN apt update && apt install htop

CMD ['sleep 1000']
```

La commande pour construire l'image est :

```
docker build [-t tag] [-f dockerfile] <build_contexte>
```

Dockerfiles



```
# our base image
FROM alpine:3.5

# Install python and pip
RUN apk add --update py2-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python
app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r
/usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html
/usr/src/app/templates/

# tell the port number the container should
expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]
```

Description du dockerfile

Instruction FROM

- L'image de base à partir de laquelle est construite l'image actuelle.

Instruction RUN

- Permet de lancer une commande shell (installation, configuration).

Instruction ADD

- Permet d'ajouter des fichier depuis le contexte de build à l'intérieur du conteneur.
- Généralement utilisé pour ajouter le code du logiciel en cours de développement et sa configuration au conteneur.

Description dockerfile

Instruction CMD

Généralement à la fin du Dockerfile : elle permet de préciser la commande par défaut lancée à la création d'une instance du conteneur avec docker run. on l'utilise avec une liste de paramètres

```
CMD ["echo 'Conteneur démarré'"]
```

Travaux-pratiques (TP1)