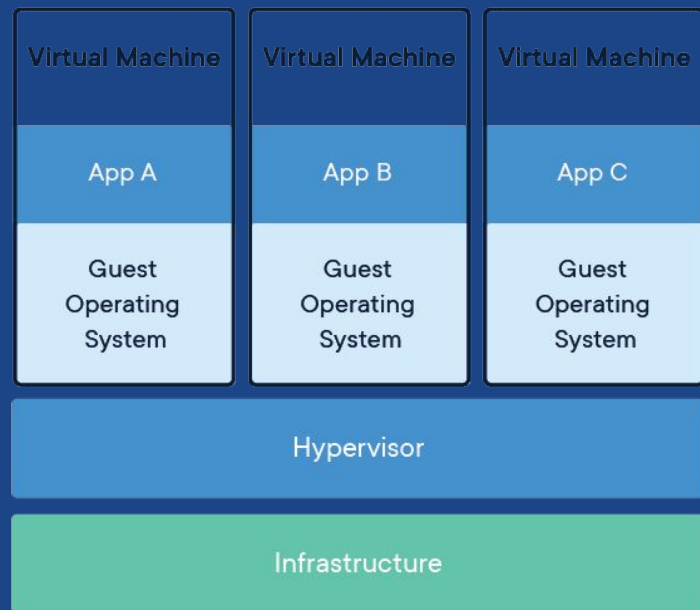
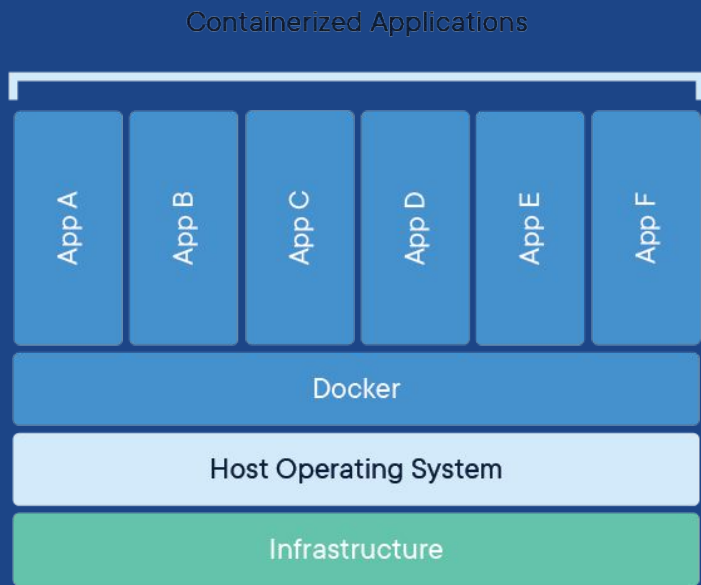


docker®

De Souza Hugo  
Delalande Alexian  
Cacheleux Vincent  
Mann Dorine

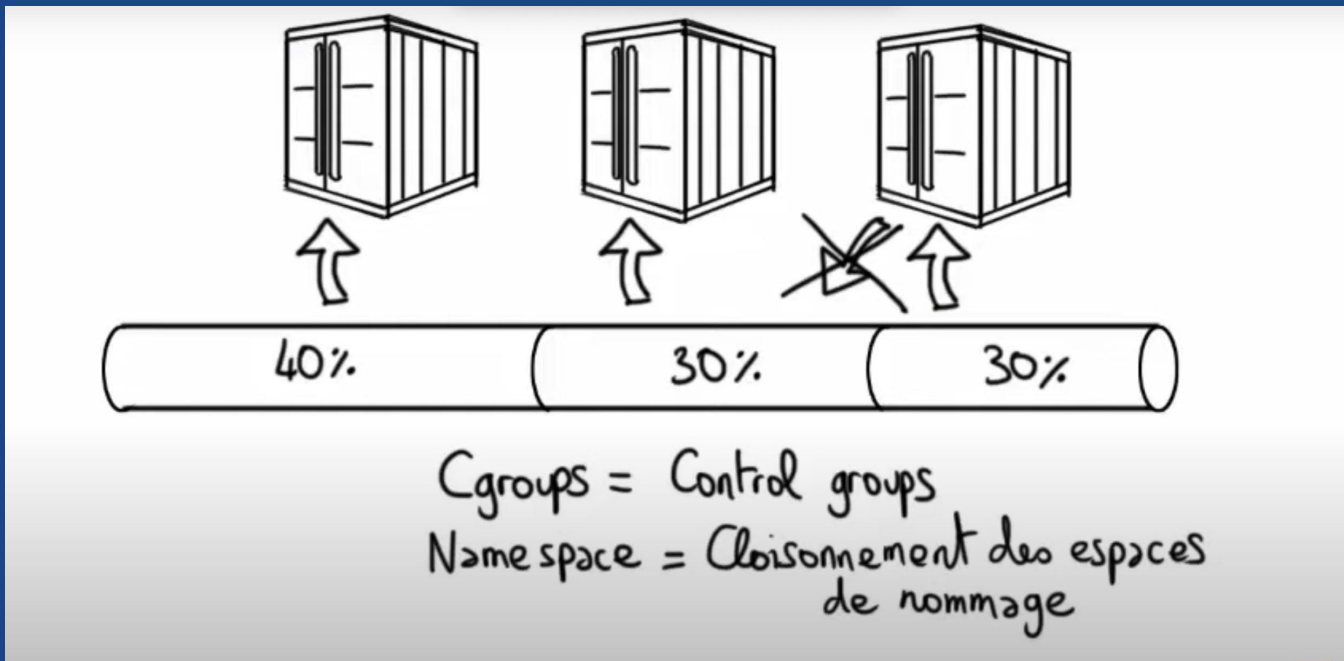
- Qu'est-ce qu'un "Conteneur" ?

- Les conteneurs se basent sur les fonctionnalités du système d'exploitation de la machine hôte
- Les conteneurs accèdent à l'OS hôte de manière totalement isolé les uns des autres
- Les conteneurs virtualisent : Le processeur, la RAM et le système de fichiers et non la machine. Ce n'est pas une VM !



- Comment ça marche ?

- Cgroups = Isoler et limiter l'utilisation des ressources
- Name space = Permet d'empêcher qu'un groupe puisse voir les ressources des autres groupes



- Avantage/ Inconvénients :

Avantages :

- Très léger
- Par conséquent facile à migrer, télécharger et installer
- Permet de développer des applications de façon plus efficiente, en utilisant moins de ressources, et de déployer ces applications plus rapidement.

Inconvénients :

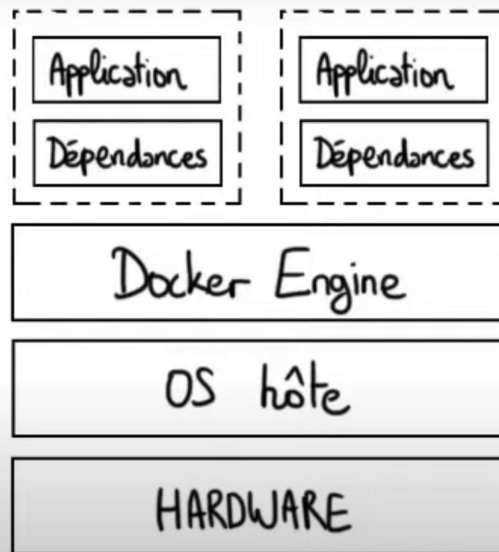
- Difficultés à gérer plusieurs conteneurs simultanéments
- Problème de sécurité : Dû au système d'exploitation hôte

- Docker

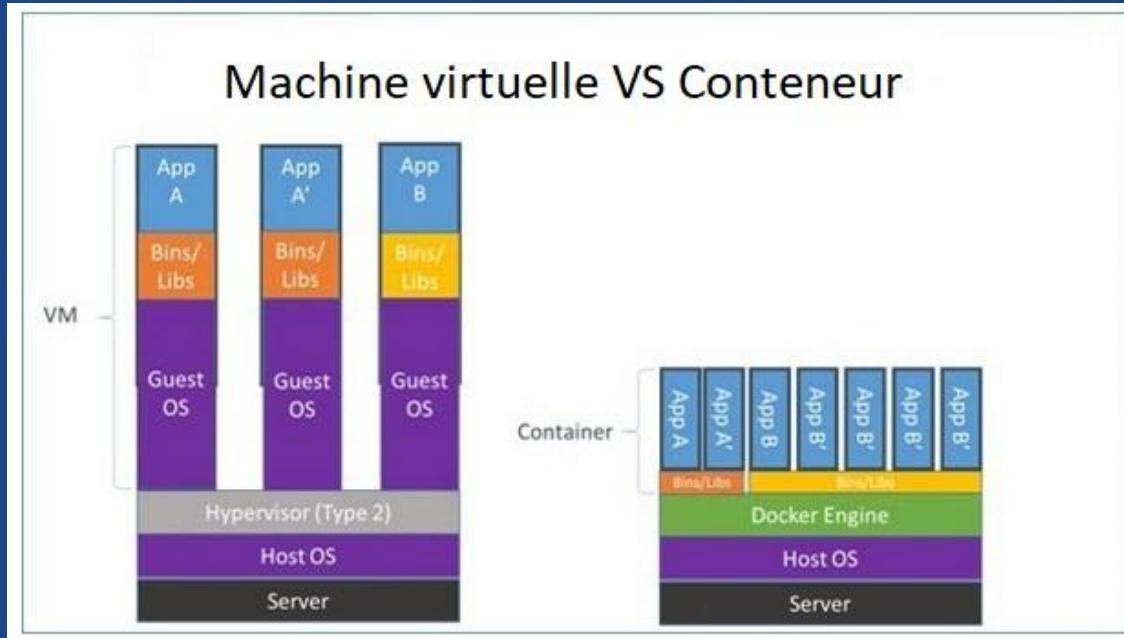
- Technologie reposant sur le LXC (Linux Containers)

- Permettant de créer des conteneurs qui vont uniquement contenir des applications avec leur dépendances.

- Les conteneurs Docker permettent d'embarquer des applications afin de les exécuter au sein de l'OS hôte mais de manière virtuellement isolé.



- Notion d'un conteneur



Cas concret d'un conteneur :

- Besoin spécifique d'une technologie complexe pour un entreprise rapidement.
- Besoin d'alléger la demande de stockage d'une entreprise grâce aux conteneurs.
- Déploiement d'une solution adapté à chaque acteurs d'une entreprise.
- Gain de temps de déploiement pour une entreprise et économie sur les moyens mis en oeuvre pour une entreprise.

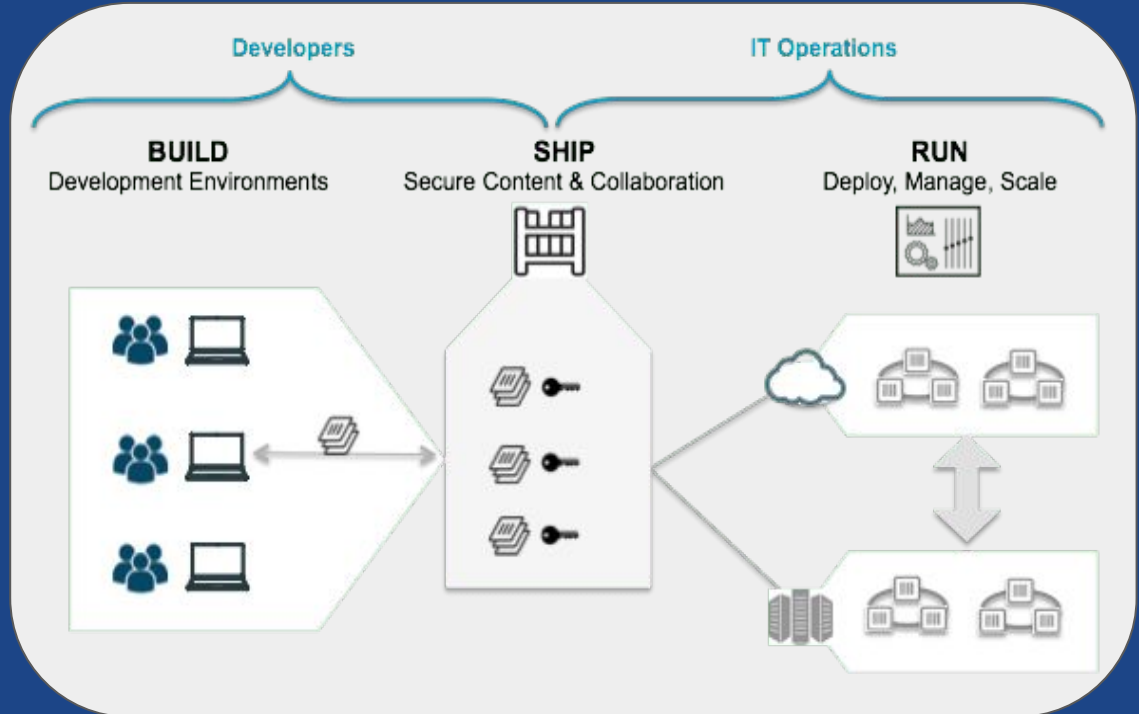
- Management de conteneur

Présentation de la solution de management “Containers as a Service” (CaaS)

Idées directrices de cette solution :

- Collaboration entre les développeurs et les opérateurs informatiques.
- Environnement de confiance avec des applications et outils sécurisé et en libre service pour le développement.
- Flexibilité des méthodes de travail pour les différents acteurs.

Diagramme d'un workflow en entreprise



- Sécurité d'un conteneur

Les conteneurs sont moins isolés les uns des autres que les machines virtuelles.





- Installation de Docker sur Debian

```
root@sv-sio-lin-pg04:/# apt update
```

```
root@sv-sio-lin-pg04:/# apt install apt-transport-https ca-certificates curl gnupg2  
software-properties-common
```



docker

```
root@sv-sio-lin-pg04:/# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add
```

```
-  
OK
```

```
root@sv-sio-lin-pg04:/# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

```
root@sv-sio-lin-pg04:/# apt update
```

```
Réception de:1 https://download.docker.com/linux/debian stretch InRelease [44,8 kB]
```

```
Réception de:2 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```



```
root@sv-sio-lin-pg04:/# apt-cache policy docker-ce
docker-ce:
  Installé : (aucun)
  Candidat : 5:19.03.9~3-0~debian-stretch
Table de version :
  5:19.03.9~3-0~debian-stretch 500
    500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
  5:19.03.8~3-0~debian-stretch 500
```

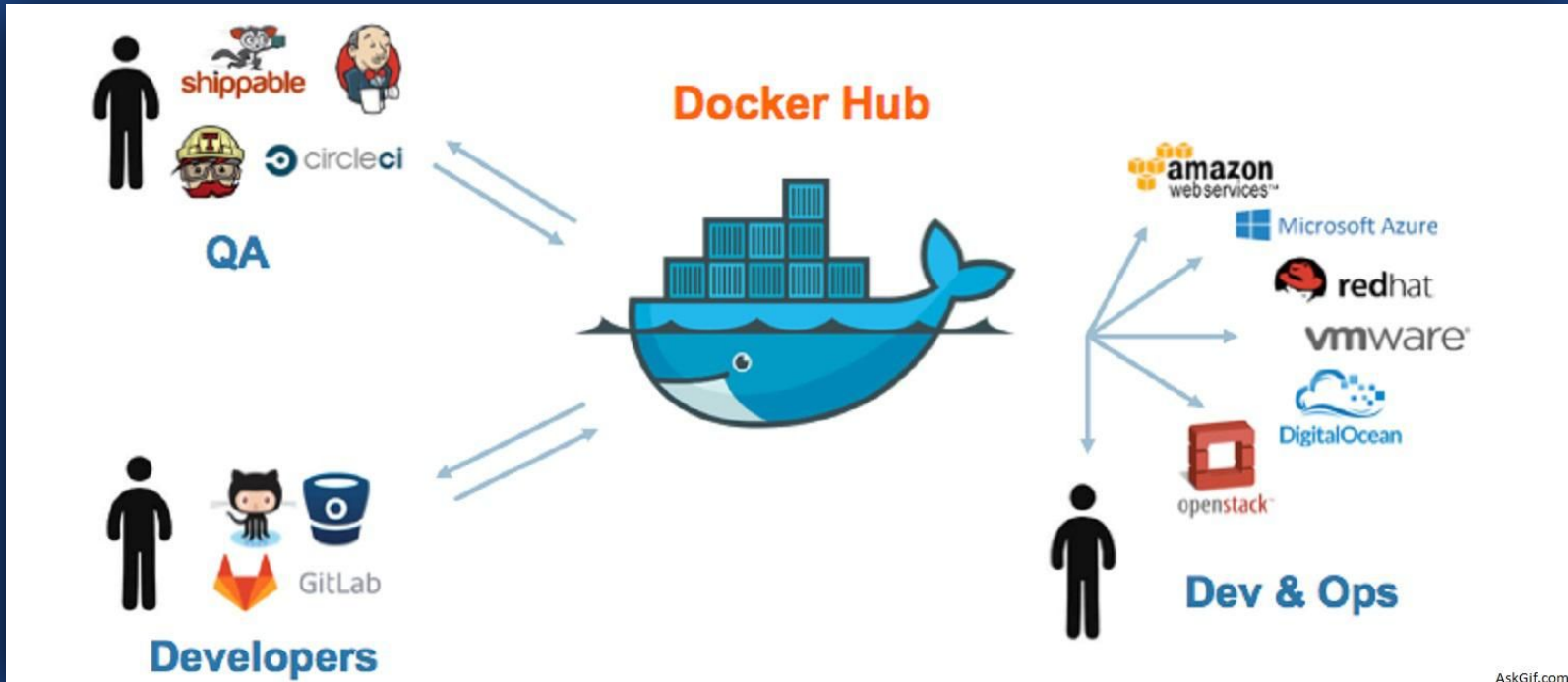
```
root@sv-sio-lin-pg04:/# apt install docker-ce
```

```
Progression : [ 31%] [#####.....]
```

```
root@sv-sio-lin-pg04:/# systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2020-05-26 17:05:41 CEST; 1min 19s ago
```

- Déploiement d'hôtes (conteneurs)

- Dockerhub est la bibliothèque de conteneurs gratuit et payant disponible pour docker.



- Déploiement à partir de Dockerhub



Commande “docker search”  
ou directement dockerhub

```
- $ docker run --name webserv -p 9001:80 -d httpd
```

```
[node1] (local) root@192.168.0.8 ~
$ docker run --name webserv -p 9001:80 -d httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
afb6ec6fdc1c: Pull complete
5a6b409207a3: Pull complete
41e5e22239e2: Pull complete
9829f70a6a6b: Pull complete
3cd774fea202: Pull complete
Digest: sha256:db9c3bca36edb5d961d70f83b13e65e552641e00a7eb80bf435cbe9912afcb1f
Status: Downloaded newer image for httpd:latest
724221c51f2f31ebc5924dd2cfc60acac07dd90bfa93c8497864188f81ef97c3
```

```
[node1] (local) root@192.168.0.8 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
724221c51f2f        httpd              "httpd-foreground" 12 minutes ago     Up 12 minutes      0.0.0.0:9001->80/tcp  webserv
```



- Création d'une image personnalisée

- Création du fichier Dockerfile dans le projet

```
FROM debian:latest
RUN apt update && apt -y install apache2 && rm -rf /var/lib/apt/lists/*
COPY ./html /var/www/html
VOLUME /var/log/apache2
EXPOSE 80
CMD ["apachectl", "-D", "FOREGROUND"]
```

- Création de l'image

```
[node1] (local) root@192.168.0.8 ~/html
$ docker build -t testing .
```

- Gestion d'un conteneur

- \$ docker exec -it <nom du conteneur> /bin/bash

```
[node1] (local) root@192.168.0.8 ~  
$ docker exec -it webserv /bin/bash  
root@724221c51f2f:/usr/local/apache2#
```

- \$ docker cp <source path> <nom du conteneur>:path

```
[node1] (local) root@192.168.0.8 ~  
$ ls  
index.html  
[node1] (local) root@192.168.0.8 ~  
$ docker cp index.html webserv:/usr/local/apache2/htdocs/index.html
```

- \$ docker stop <nom du conteneur>
- \$ docker start <nom du conteneur>
- \$ docker rm <nom du conteneur> ##Le conteneur doit être à l'arrêt.
- \$ docker stats
- \$ docker ps