



# INTRODUCTION AUX SYSTÈMES DISTRIBUÉS

INF36307 – SYSTÈMES DISTRIBUÉS

# AGENDA – COURS 2 – LES CONTENEURS

1

## Images & Conteneurs

Le site web de l'UQAR

2

## Volume

Gestion du stockage

3

## Réseautique

Communication inter  
conteneurs

4

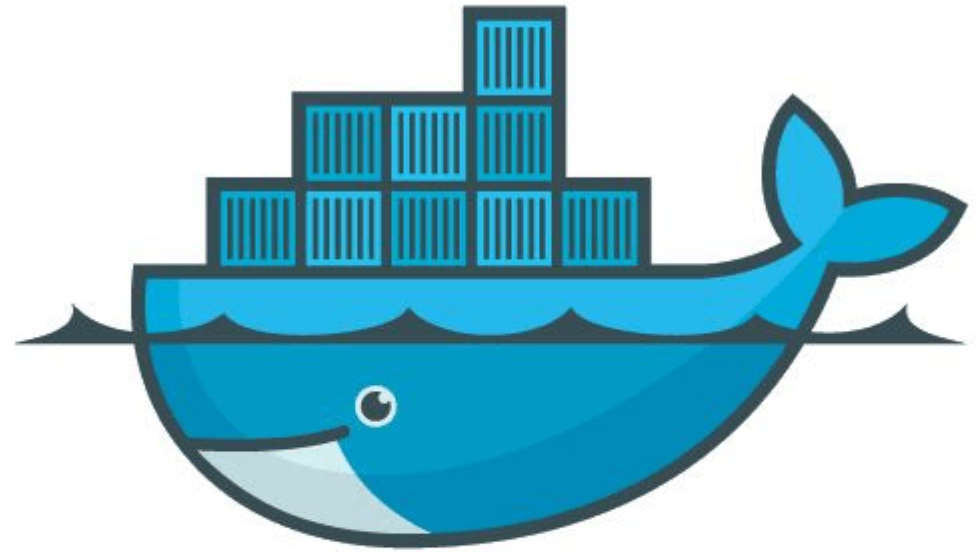
## Composition

Créer un amalgame de  
conteneur

---

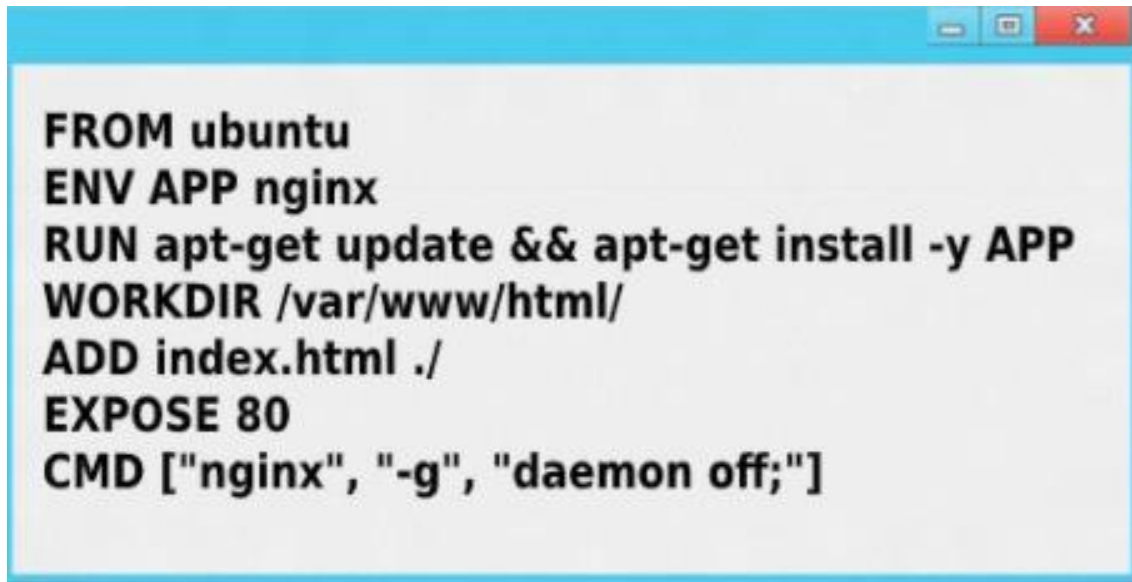
# DOCKER

- Une plateforme Open Source
- Facilite le développement & déploiement de systèmes
- Encapsulation de l'environnement
- N'est pas une machine virtuelle



# docker

# DOCKER IMAGE

A screenshot of a Dockerfile, which is a text file used to define how to build a Docker image. The file contains several instructions: 'FROM ubuntu' specifies the base image; 'ENV APP nginx' sets an environment variable; 'RUN apt-get update && apt-get install -y APP' installs the application; 'WORKDIR /var/www/html/' sets the working directory; 'ADD index.html .' adds a file; 'EXPOSE 80' indicates the port the container listens on; and 'CMD ["nginx", "-g", "daemon off;"]' sets the command to run when the container starts.

```
FROM ubuntu
ENV APP nginx
RUN apt-get update && apt-get install -y APP
WORKDIR /var/www/html/
ADD index.html ./
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- Une copie reproductible d'un système
- Décrite avec un langage de template
  - Configurée dans un fichier Dockerfile
- Qui contient une application et ses dépendances
  - Ex: librairies, outils, ping,
- Permet d'hériter d'une autre images
- Chaque instruction est une couche (« layer ») qui peut être mise en cache.
- Peut définir des variables d'environnement et une command à exécuter

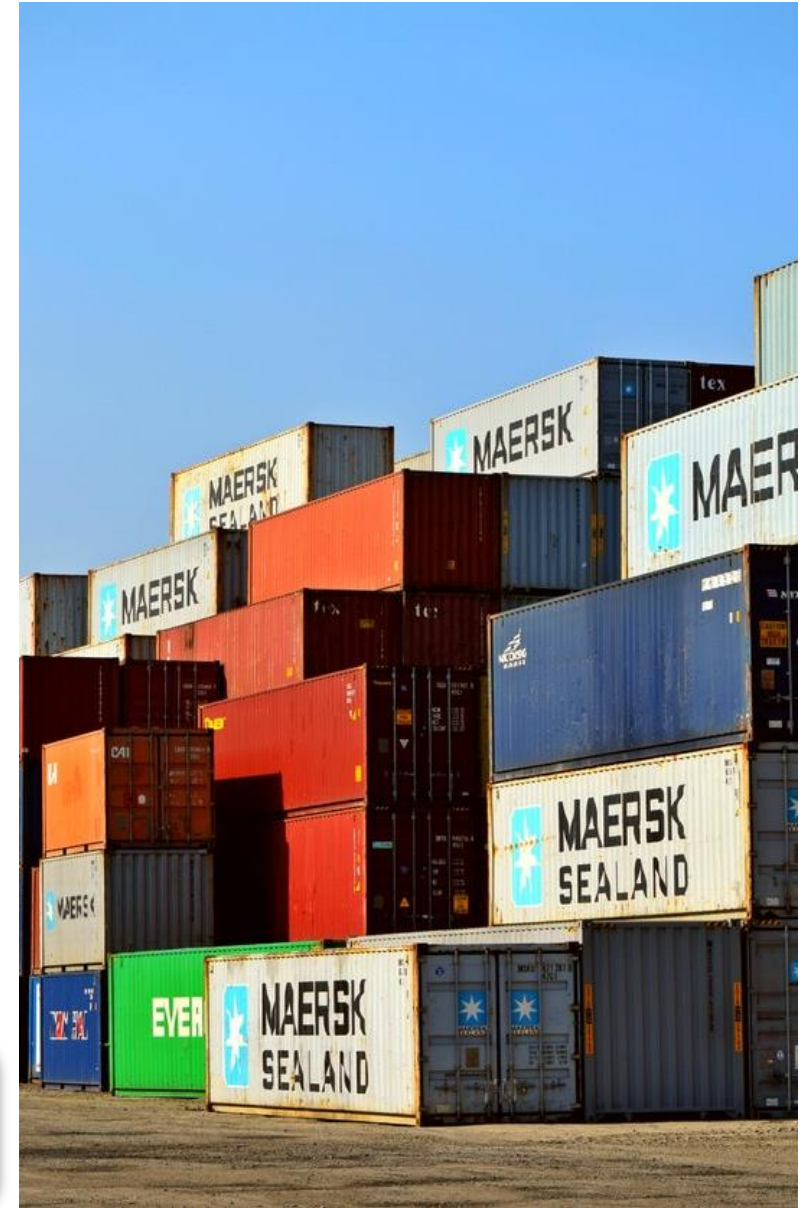
Business Continuity: Rouler plus d'un processus dans un seul conteneur -> systemd

# DOCKER CONTENEUR

- Un process sur une machine, isolé des autres process
- Utilise les notions de Linux Namespace pour l'isolation
- Utilise `chroot` pour simuler un sous-espace de fichier indépendant
- Utilise des groupe de contrôles (`cgroup`) pour limiter les ressource disponible pour un conteneur

OOM kill: la gestion de la mémoire

Héberger le site web de l'UQAR localement dans un conteneur.





# IMAGE REGISTRY

- Les images dockers sont facilement partageables via un docker registry
- Une fois dans le registry l'image peut être exécuté de n'importe où.
- C'est la façon traditionnelle de déployer en production une application dans un conteneur



Créer une image et la partager via dockerhub

# PROGRESSION

Images et conteneurs

Stockage

Réseautique

Composition





## GESTION DU STOCKAGE

- De base dans un conteneur, tous changements de fichiers sur le disque seront perdu à la fin de vie du conteneur
- Si la persistance de donnée est requise, docker offre la notion de volume
- Les volumes sont des unités de stockage que l'on attache à un conteneur
- Il existe 2 types de volume
  - Named volume
  - Bind Mount

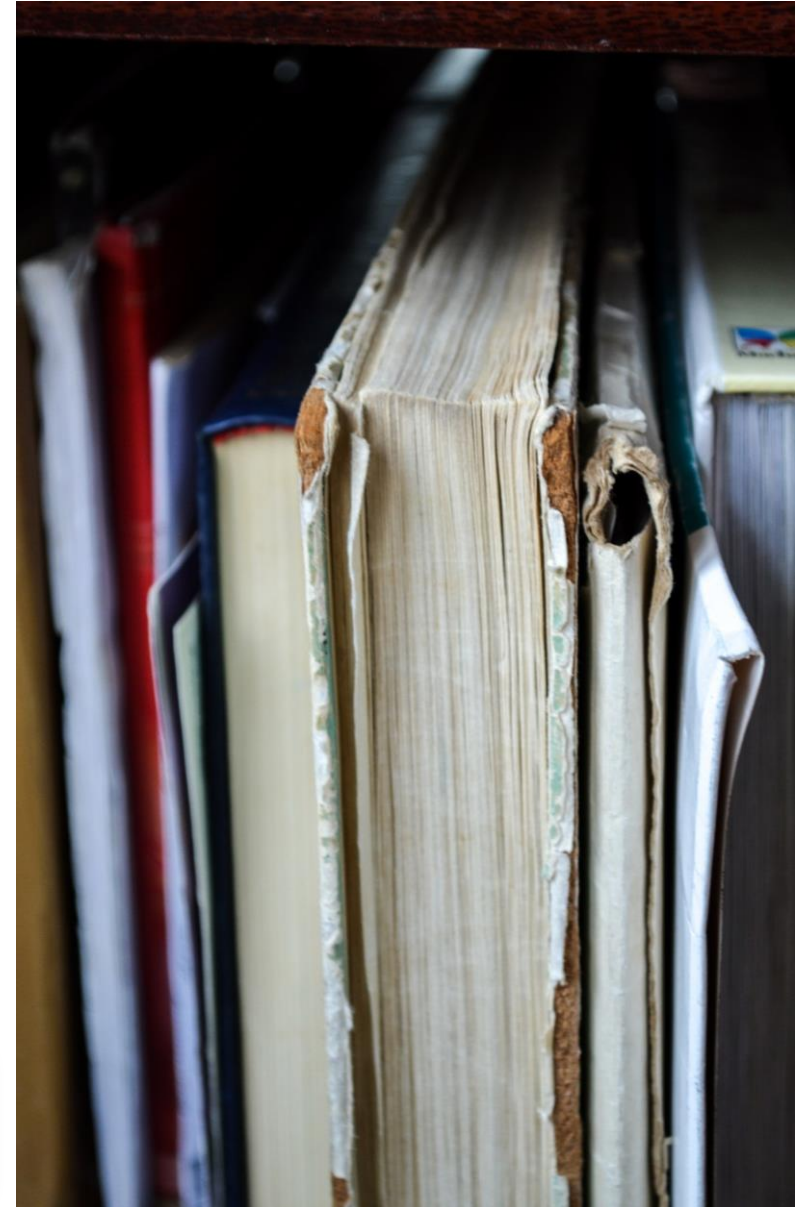


## BIND MOUNT

- Surtout utilisé lors du développement local
- Permet d'avoir un répertoire de la machine hôte exposé dans un conteneur

```
docker run -it --rm -p 80:80 -v  
"$ (pwd) /files:/usr/share/nginx/html" cours2-siteweb
```

Modifier dynamiquement un site web sans relancer de conteneur

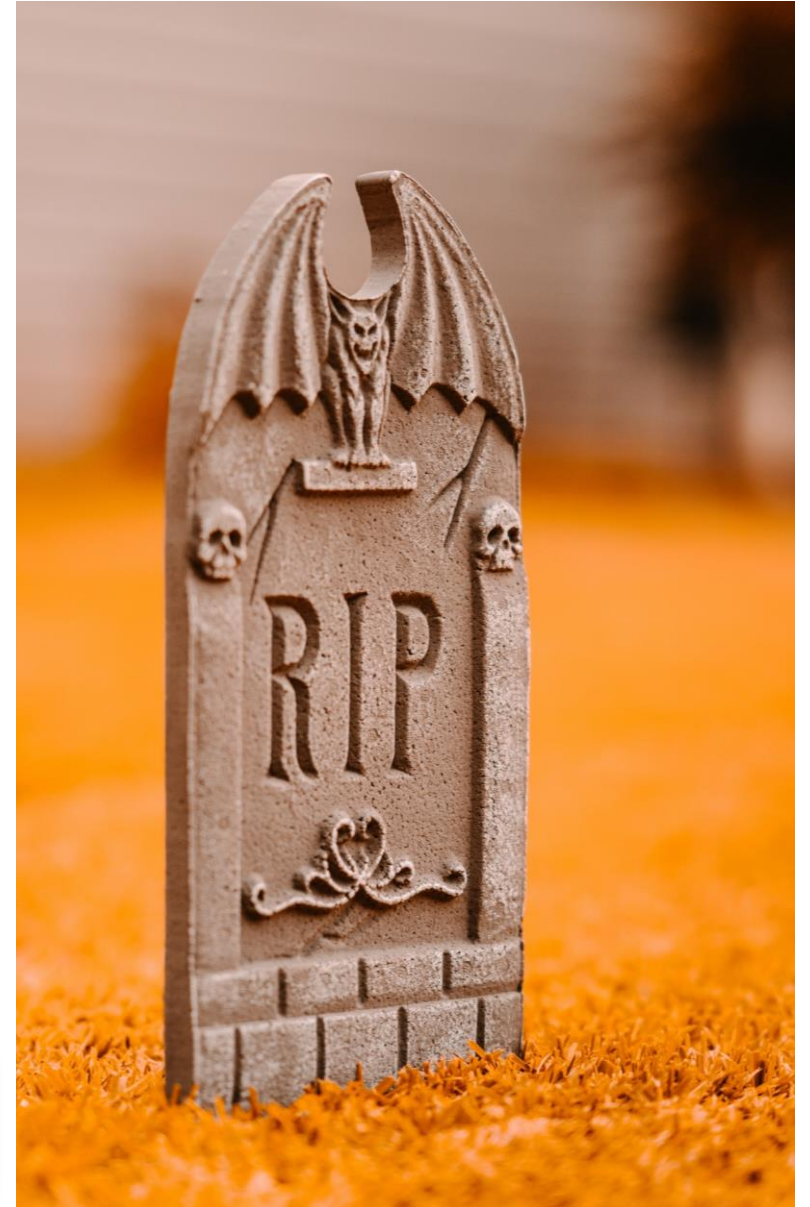


## NAMED VOLUME

- Surtout utilisé pour partager des données entre plusieurs conteneurs
  - En production
- Permet d'attacher un répertoire à un conteneur
- Ce type de volume persiste après la destruction d'un conteneur

```
docker volume create cours2  
docker run -it --rm -p 80:80 -v  
"cours2:/usr/share/nginx/html" cours2-siteweb
```

Incrémenter le nombre d'étudiants sur le disque et persister l'information.



# LES VOLUME SPÉCIAUX

- Permet de stocker les données dans des services externes dans le cloud
- Principalement utilisés lorsque le volume doit suivre le conteneur sur différentes machines physiques
- Très peu utilisés
- Exemples: AWS S3, Azure, flocker



# PROGRESSION

Images et conteneurs

Stockage

Réseautique

Composition





# RÉSEAUTIQUE DE CONTENEUR

- Chaque conteneur est isolé dans son propre réseau et n'a pas de visibilité sur les autres conteneurs sur la machine hôte
- Pour permettre à des conteneurs de communiquer entre eux, un réseau doit être créé:

```
docker network create cours2
```

- Le réseau doit être attaché au conteneur lors de l'exécution:

```
docker run --network cours2 --network-alias mysql
```

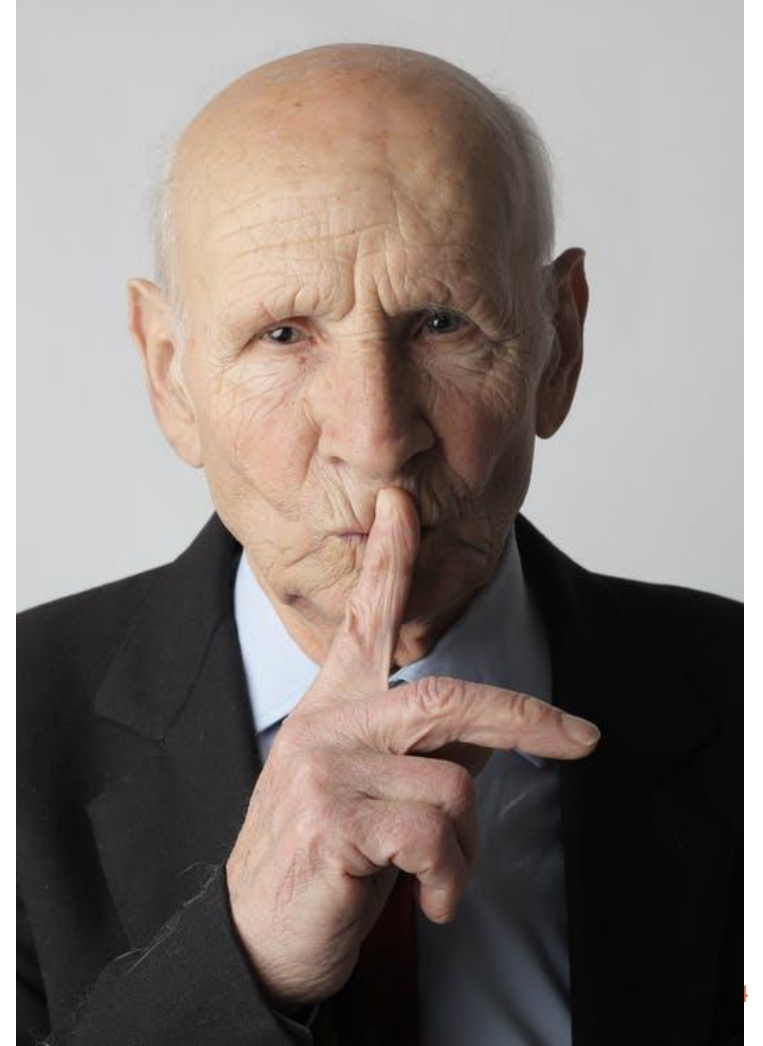
- `--network-alias` permet à un conteneur d'être identifié par nom au lieu d'adresse IP





# LES SECRETS

- Permet de rendre disponible au conteneur des informations sensibles (ex: password) sans les rendre visibles dans les fichiers de configuration
- `docker secret create`





## ÉTUDE DE CAS – AFFICHER LA LISTE DES ÉTUDIANTS

- Créer une base de donnée (MYSQL)
- Créer un web service (go)
- Afficher les données de la base de donnée au format JSON depuis le webservice

# PROGRESSION

Images et conteneurs

Stockage

Réseautique

Composition



# DOCKER COMPOSE

- Est un outil qui permet de configurer et de lancer plusieurs conteneurs
- Un fichier `docker-compose.yml` permet de définir la configuration de l'ensemble des conteneurs
- Un docker network est créé par défaut pour la communication entre les conteneurs

```
docker-compose build
docker-compose up
docker-compose down
```

Convertir votre web service en format docker-compose.

Usine à gaz: Tester localement avec toutes les dépendances





## CONSEILS DE PRO

- Optimiser la taille de l'image docker
  - `.dockerignore`
  - Images intermédiaires

```
FROM golang:1.19 as builder
...
COPY --from=builder /go/cmd/main .
CMD ["/main"]
```





# DEVOIR

ÊTRE EN MESURE D'UTILISER KUBERNETES (MINIKUBE) SUR VOTRE ORDINATEUR PERSONNEL

```
minikube start
```

```
minikube dashboard
```