

# Dockerfile

\* commentaire

LTS

SSH

**FROM** nom l'image docker + version  
↳ (placement en dessous)

**ENV** MYSQL\_ROOT\_PASSWORD=azerty

**ENV** MYSQL\_DATABASE=mydatabase

**ENV** MYSQL\_USER=mike

**ENV** MYSQL\_PASSWORD=1234

**RUN** apt-get update

**RUN** apt-get install -y --fix-missing /

openssh-client openssh-server

docker build -t postgres:2503

docker images

-d

docker run -name postgres-1 -p 4003:5432 -d postgres:2503

# Gestion des Images

## Docker

\* Construire une image

docker build -t monimage

\* Affiche toutes les images

docker images

docker images monimage

\* Télécharge une image depuis un registre

docker pull mysql:5.7

\* Pousse une image vers un registre

docker push mysql:5.7

\* Supprime une ou plusieurs images

docker rmi monimage

\* Crée un nouveau tag pour une image existante

docker tag monimage:1 monimage:26

# Gestion des Conteneurs

dt Crée et démarre un nouveau conteneur

docker run -d -p Port:Port M : C -name C -v img:/var/lib/docker/images

dt Démarrage d'un conteneur

docker start monconteneur ou ID

dt Arrête un conteneur

docker stop monconteneur ou ID

dt Redémarre un conteneur

docker restart monconteneur ou ID

docker restart --time 10 monconteneur

dt Affiche les conteneurs en cours d'exécutions

docker ps

docker ps -a // --all Tous C même si ét

docker ps -d // --quiet Affichage silencieux

docker --format "table {{.Name}}{{\t}}{{.Status}}{{\t}}{{.Ports}}

dt Supprime un ou plusieurs conteneurs

docker rm monconteneur ou ID

docker rm --force monconteneur

## Interaction avec les Conteneurs

\* Lance une commande dans un conteneur  
`docker exec -it monconteneur /bin/bash`

\* Affiche les logs d'un conteneur  
`docker logs monconteneur`

\* Inspecter un Conteneur

`docker inspect monconteneur`

\* Crée, Affiche, Supprime, Inspecte

`docker volume create nomvolume` //Crée

`docker volume ls` //affiche

`docker volume rm` //Supprime un ou plusieurs

`docker volume inspect` //inspecter

## Nettoyage et Maintenance

# Supprime ce qui est inutilisé

docker image prune -a // extension possible  
docker container prune  
docker volume prune  
docker system prune // -a, --volume, -force

## Information système

# Affiche info système, version, stats, processus

docker info // --format '{\"ServerVersion\":\"{{.ServerVersion}}\"}'  
docker version?

docker top nonconteneur // processus

docker stats // non conteneur

// --format "table {{.Containers}}\t{{.CPUPerc}}\t{{.Memory}}

# Docker Compose

# Démarrer tous les services docker-compose.yml  
docker-compose up // -d // --build // service1 // service2 // scale

docker-compose down // arrête // rm il alle supprime web-3

docker-compose build // service1 // --full // construit

docker-compose ps // --services // affiche l'état

# Crée un réseau Gestion Réseaux

docker network create nonreseau

docker network create --driver bridge nonreseau

Exemple

docker network create --subnet 172.20.0.0/16 nonreseau

# Affiche réseaux

docker network ls

docker network ls --filter driver=bridge

docker network ls --format "table {{.Name}}\t{{.Driver}}

# Connecte un conteneur à un réseau

docker network connect nonreseau nonconteneur

docker network connect --ip 172.20.0.10 nonreseau nonconteneur

# Déconnecte un Conteneur d'un réseau

docker network disconnect nonreseau nonconteneur

docker network disconnect --force nonreseau nonconteneur

## Registre et Authentification

# Se connecter au registre Docker

docker login //--password-stdin registry.com

docker login registry.com

docker login -u utilisateur registry.com

# Se déconnecter du registre Docker

docker logout

( docker logout registry.com

# Rechercher une image sur Docker Hub

docker search nginx

! docker search --limit 10 ubuntu

.c docker search --filter stars=100 python

## Commande Avancées

# Exporter ou Importer des images sous forme d'archive  
docker commit nonconteneur newimage

docker commit -m "message commit" nonconteneur newimage; vi

docker commit --author "nom<email>" nonconteneur newimage

# Exporte ou Importe des images sous forme d'archive

docker save nomimage > nomimage.tar

docker save -o nomimage.tar nomimage; vi

docker load < nomimage.tar

docker load -i nomimage.tar

# Exporte ou Importe des conteneurs

docker export nomconteneur > nomconteneur.tar

docker export -o nomconteneur.tar nomconteneur

docker import conteneur.tar nouvelleimage:imported

- P préciser le Fichier Hash  
patch (Mageur)   
compiler ~~Kali~~ une mysql : 8.4.6   
version latest pour la prod ajout de nouvelle  
Simon toujours prendre une fonctionnalité  
version stable avant latest mineur

- d, --detach : Mode arrière-plan
- it : Mode interactif avec le Terminal et le conteneur
- p, --publish : Mappage de ports (machine:conteneur)
- v, --volume : Mountage de volumes
- e, --env : Variable d'environnement
- name : Nommer Conteneur
- rm : Supprime Conteneur --rmi Supprime Image
- f, --Force : Forcer opération
- a, --all : inclus tous (même arrêtés)
- q, --quiet : Affichage silencieux (ID seulement)