

## Commande utile pour docker

#### DÉMARRAGE / MISE HORS TENSION

```
$ systemctl start docker
$ systemctl stop docker
### LANCER UNE IMAGE AVEC UBUNTU ###
```

```
$ docker run -it ubuntu bash
### ARRÊTER UNE IMAGE ###
```

```
$ docker rm c0bdacb3478b
c0bdacb3478b correspond à l'ID de l'image
### LISTE DE MES IMAGES ###
```

```
- Toutes les images:
$ docker images
- Les images actives:
$ docker ps
### CRÉATION D'UNE IMAGE ###
```

```
- Créer un Dockerfile
/*
FROM ubuntu:latest
MAINTAINER jgal
RUN apt-get update \
&& apt-get install -y vim git \
&& apt-get clean \
&& rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
*/
```

```
- Puis construire l'image
$ docker build -t premiereImage:v1.0 .
- Dans ce cas lancer
$ docker run -it premiereImage:v1.0 bash
### HISTORIQUE D'UNE IMAGE ###
```

```
$ docker history premiereImage:v1.0
### CREER UN CONTAINER DEPUIS NOTRE IMAGE ###
```

```
$ docker run -tid --name test premiereImage:v1.0
- C'èr un container avec un port
$ docker run -tid -p 8080:80 --name web nginx:latest
### SUPPRIMER UN CONTAINER ###
```

```
$ docker rm -f test
### SUPPRIMER UNE IMAGE ###
$ docker rmi premiereImage:V1.0
### DEMMARER / ARRÊTER UNE IMAGE ###
```

```
- Démarrer
$ docker start nomImage
- Arrêter
$ docker stop nomImage
### ACCÉDER AU SHEL DE L'IMAGE ###
```

```
$ docker exec -ti nomImage sh
### CREER UN VOLUME LIER A UN CONTAINER ###
```

```
$ docker run -tid -p 8080:80 -v /chemin/dans/locale:/chemin/dans/container --name web nginx:latest
```