



De la découverte à l'expertise

DOCKER CHEAT SHEET

REGISTRE (DOCKER HUB, ...)

Recherche une image dans le registre

```
# docker search <image_name>
```

Rapatrier une image du registre

```
# docker pull <image_name>
```

Se connecter au registre

```
# docker login -u <user_name>
```

Publier une image sur le registre

```
# docker push <user_name>/<image_name>
```

IMAGES

Lister les images locales

```
# docker images OU # docker image ls
```

Supprimer une image locale

```
# docker rmi <image_name>
```

Supprimer toutes les images non utilisées

```
# docker image prune
```

Construire une image à partir d'un DOCKERFILE

```
# docker build -t <image_name> <DIRECTORY>
```

CONTENEURS

Créer et démarrer un conteneur en précisant son nom :

```
# docker run --name <container_name> <image_name>
```

Démarrer un conteneur en publiant un port :

```
# docker run -p <host_port>:<container_port>  
<image_name>
```

Démarrer un conteneur en publiant tous les ports :

```
# docker run -P <image_name>
```

Démarrer un conteneur en tâche de fond (daemon)

```
# docker run -d <image_name>
```

Démarrer ou arrêter un conteneur existant:

```
# docker start|stop <container_name> (or  
<container_id>)
```

Supprimer un conteneur arrêté :

```
# docker rm <container_name>
```

Ouvrir un Shell à l'intérieur d'un conteneur :

```
# docker exec -it <container_name> <executable>
```

Suivre les logs d'un conteneur :

```
# docker logs -f <container_name>
```

Afficher les détails d'un conteneur:

```
# docker inspect <container_name> (or  
<container_id> )
```

Lister les conteneurs en cours d'exécution:

```
# docker ps
```

Lister tous les conteneurs (démarrés et arrêtés):

```
# docker ps -a
```

Afficher l'utilisation des ressources :

```
# docker stats
```

VOLUMES

Lister les volumes :

```
# docker volume ls
```

Créer un volumes:

```
# docker volume create <volume_name>
```

RESEAUX

Lister les volumes :

```
# docker network ls
```

Créer un volumes:

```
# docker network create <volume_name>
```

ATTENTION

- Il est important de maîtriser l'option « **inspect** » qui peut être utilisé dans presque tous les commandes. Elle facilitera vos opérations de debug.

- Il faut faire attention lors de l'utilisation de la commande « **prune** » qui permet de faire le ménage.