



Docker Cheat Sheet

Stéphane ROBERT



Qu'est-ce que Docker ?



Docker est une plateforme open-source conçue pour automatiser le déploiement d'applications dans des **conteneurs logiciels**. Ces conteneurs permettent aux développeurs d'emballer une application avec toutes ses dépendances afin qu'elle fonctionne de manière uniforme et cohérente sur différentes machines.

Fonctionnalités Clés

- **Portabilité** : Fonctionne de manière identique peu importe l'environnement (dev, test, production).
- **Isolation** : Chaque conteneur est isolé des autres, améliorant la sécurité et la stabilité.
- **Efficacité** : Démarrage rapide des conteneurs et meilleure utilisation des ressources.

Installation et Configuration

- Dans une fenêtre Terminal, tapez les commandes suivantes :

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
$ sudo usermod -aG docker ${USER}
```

- Vérifier que Docker CLI est installé

```
$ docker --version
```

Commandes Générales

- Démarrer Docker

```
$ sudo systemctl start docker
```

- Vérifier le statut de Docker

```
$ sudo systemctl status docker
```

Commande Docker Run

```
$ docker run <options> <image_name>
```

Exemples pratiques :

- Lancer un conteneur en mode interactif avec un terminal bash :

```
$ docker run -it <image_name> /bin/bash
```

- Lancer un conteneur en arrière-plan (détaché) :

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

- Lancer un conteneur et mapper un port local avec celui du conteneur :

```
$ docker run -d -p 8080:80 <image_name>
```

- Monter un volume dans le conteneur :

```
$ docker run -v /local/path:/container/path <image_name>
```

- Nommer un conteneur lors du lancement :

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

Gestion des Conteneurs

- Lister tous les conteneurs

```
$ docker ps -a
```

- Démarrer un conteneur existant

```
$ docker start <container-id>
```

- Arrêter un conteneur

```
$ docker stop <container-id>
```

- Supprimer un conteneur

```
$ docker rm <container-id>
```


Gestion des Images Docker

- Lister toutes les images Docker locales

```
$ docker images
```

- Télécharger une image Docker (pull)

```
$ docker pull <image-name>
```

- Construire une image Docker à partir d'un Dockerfile

```
$ docker build -t <image-name> .
```

- Supprimer une image Docker

```
$ docker rmi <image-id>
```

Gestion des Volumes Docker

- Lister tous les volumes

```
$ docker volume ls
```

- Créer un volume

```
$ docker volume create <volume-name>
```

- Supprimer un volume

```
$ docker volume rm <volume-name>
```

Surveillance des Événements Docker

Surveillez les événements en temps réel sur le démon Docker, tels que le démarrage ou l'arrêt des conteneurs.

- Utiliser la commande `docker events` :

```
$ docker events
```

Cette commande affiche les événements système Docker en temps réel, comme :

- Le démarrage et l'arrêt des conteneurs
- Les changements d'état des volumes et des réseaux
- Les mises à jour des images Docker

Nettoyage des Ressources avec Docker Prune

- Nettoyer les conteneurs, images et volumes non utilisés

```
$ docker system prune
```

Cette commande supprime :

- Les conteneurs arrêtés
- Les réseaux non utilisés
- Les images dangling (celles qui ne sont pas taggées ou associées à un conteneur)
- Les caches de build non utilisés
- Pour nettoyer les volumes également :

```
$ docker system prune --volumes
```

- Pour forcer la suppression sans confirmation (non interactif) :

```
$ docker system prune -f
```

- Pour un nettoyage ciblé :

```
$ docker container prune
```

- Supprimer uniquement les images non utilisées :

```
$ docker image prune
```

- Supprimer uniquement les volumes inutilisés :

```
$ docker volume prune
```

Plus loin avec Docker

Avec ces connaissances et astuces, tu es désormais mieux équipé pour gérer tes conteneurs Docker de manière efficace et fluide.

Si vous souhaitez apprendre plus de choses sur Docker je vous invite à vous rendre sur mon site.

<https://blog.stephane-robert.info/docs/conteneurs/moteurs-conteneurs/docker/>