

Documentation Docker



SOMMAIRE

- **Installation de Docker** 3
- **Vérification du bon fonctionnement de Docker** 4
- **Vérifier et gérer le statut de Docker** 4
- **Gérer les images** 5
- **Gérer les conteneurs** 5
- **Créer un conteneur personnalisé** 6
- **Rendre un service accessible** 6
- **Problème avec “usermod”** 7



Installation de Docker

Tout d'abord, on va commencer par installer les outils nécessaires :

- `apt install -y ca-certificates curl gnupg lsb-release`
- `apt install curl`

Ensuite on va ajouter la **clé GPG** officielle de Docker avec la commande :

- `mkdir -p /etc/apt/keyrings`
- `curl -fsSL https://download.docker.com/linux/debian/gpg`
- `sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`

Ajout du dépôt Docker avec la commande :

- `echo \ "deb [arch=$(dpkg --print-architecture)
signedby=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \ $(lsb_release -cs)
stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null`

Installation de Docker Engine :

- `apt install -y docker-ce docker-ce-cli containerd.io docker-buildx
plugin docker-compose-plugin`

Puis ajout de l'utilisateur au groupe Docker :

- `apt install -y passw`
- `sudo usermod -aG docker $USER`

Vérification du bon fonctionnement de Docker

Vérifier la version de Docker installée :

- `docker --version`

Puis tester le fonctionnement de Docker

- `docker run hello-world`

Vérifier et gérer le statut de Docker

Consulter le statut du service Docker :

- `systemctl start docker`
- `systemctl enable docker`
- `systemctl restart docker`
- `systemctl stop docker`

Et vérifier que Docker soit bien actif :

- `systemctl status docker`

Gérer les images

Rechercher une image (Docker Hub) :

- `docker search "nom_image"`

Télécharger une image :

- `docker pull "nom_image"`

Puis lister les images locales :

- `docker images`

Enfin supprimer une image :

- `docker rmi "image_id"`

Gérer les conteneurs

Tout d'abord créer et démarrer un conteneur :

- `docker run "nom_image"`

Puis lister les conteneurs actifs :

- `docker ps -a`

Pour supprimer un conteneur arrêté :

- `docker rm "conteneur_id"`

Démarrer un conteneur existant :

- `docker start "conteneur_id"`

Et enfin pour renommer un conteneur :

- `docker rename ancien_conteneur_name nouveau_conteneur_name`

Et pour finir , bien vérifier le changement avec la commande :

- `docker ps-a`

Créer un conteneur personnalisé

Tout d'abord créer et démarrer un conteneur en interactif :

- `docker run -it --name ubuntu /bin/bash`

Puis personnaliser le conteneur avec la commande :

- `apt update && apt install -y nano`

Puis sauvegarder l'état du conteneur sous forme d'image :

- `docker commit "nom_conteneur" "nom_image"`
- `docker commit`

Enfin vérifier que l'image est disponible :

- `docker images`

Rendre un service accessible

Tout d'abord lancer un conteneur avec un port exposé :

- `docker run -d -p "port local" : "port conteneur" "nom_image"`

Petit exemple avec SSH avec la même commande :

- `docker run -d -p 2222:22 ubuntu:latest`

Ensuite vérifier le mappage des ports :

- `docker ps`

Enfin tester l'accès au service , on va prendre pour exemple SSH comme service :

- `ssh root@"ip_locale" -p 2222`

Problème avec “usermod”

Tout d’abord installer “usermod” avec les commandes :

- `apt update`
- `apt install -y passwd`

Puis ensuite ajouter l’utilisateur au groupe Docker avec la commande :

- `sudo usermod -aG docker $USER`

Et si “usermod” n’est toujours pas disponible , il faudra utiliser la commande suivante :

- `sudo gpasswd -a $USER docker`

Et pour finir il faut appliquer tous ces changements avec la commande :

- `newgrp docker`

ou alors :

- `reboot`

