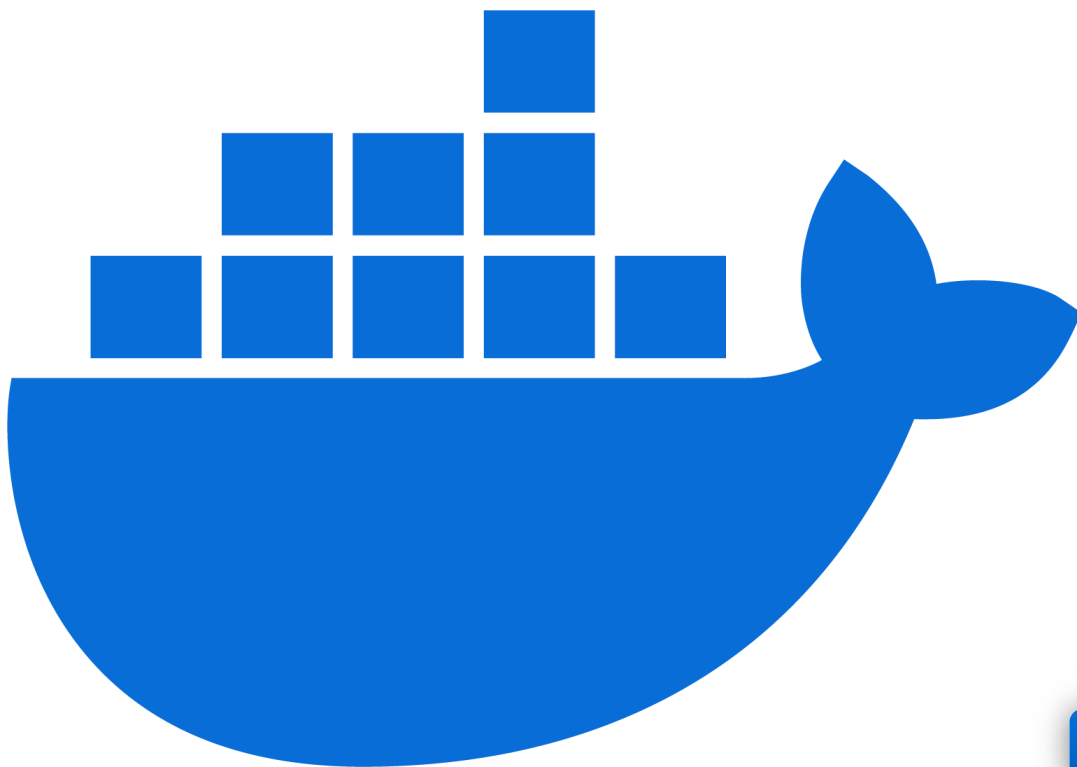


# Mémento

## Docker

---

Version 1.0 (créé le 30/10/2024, modifié le 30/10/2024)



Description



**Loric Informatique**

# 1. Installations

## 1.1. Installer Docker

Sur Windows, plusieurs solutions s'offrent à vous : Soit vous téléchargez Docker sur une machine virtuelle Linux, soit vous installez Docker Desktop pour Windows (AMD64 et non ARM64).

Dans le terminal Linux, entrez la commande :

```
sudo apt install docker.io
```

La commande `docker -v` permet de vérifier la présence de Docker.

## 1.2. Installer Docker-Compose

Dans le terminal Linux, entrez la commande :

```
sudo apt install docker-compose
```

Ou (si distutils n'a pas été installé correctement) :

```
sudo apt install docker-compose python3 python3-setuptools
```

La commande `docker-compose -v` permet de vérifier la présence de Docker-Compose.

# 2. Docker

## 2.1. Les commandes Docker de base

Les commandes ci-dessous nécessitent de privilèges administrateurs (avec `sudo` avant la commande).

Astuce : Pour ne pas avoir à utiliser tout le temps `sudo`, il est possible d'activer un compte root en définissant un mot de passe avec `sudo passwd root`, et d'ouvrir un terminal avec la commande : `su`

## 2.1.1. Les images Docker

Les images permettent de créer des conteneurs à partir de modèles déjà existants.

### 2.1.1.1. Créer une image Docker

Instruction	Description
<code>docker build -t monimage ./</code>	Créer une image

### 2.1.1.2. Afficher les images Docker

Instruction	Description
<code>docker images</code> ou <code>docker image ls</code>	Afficher les différentes images
<code>docker images -a</code>	Afficher les différentes images et les images intermédiaires

### 2.1.1.3. Supprimer une image Docker

Instruction	Description
<code>docker rm monimage</code> ou <code>docker image rm monimage</code>	Supprimer une image
<code>docker system prune</code>	Supprimer le cache, les conteneurs arrêtés et les images sans nom

### 2.1.1.1. Autres commandes Docker

Instruction	Description
<code>docker pull monimage</code>	Télécharger une image

## 2.1.2. Les conteneurs Docker

Les conteneurs permettent d'exécuter un environnement isolé du système d'exploitation à partir d'images existantes.

### 2.1.2.1. Créer un conteneur Docker

Instruction	Description
<code>docker run monimage</code>	Démarrer un conteneur à partir d'une image
<code>docker run --name monconteneur monimage</code>	Démarrer un conteneur à partir d'une image et choisir le nom du conteneur
<code>docker run -it monimage</code>	Démarrer un conteneur en mode interactif (utile pour des commandes nécessitant des entrées dans la console, comme bash)
<code>docker run -rm monimage</code>	Démarrer un conteneur et le supprimer automatiquement lorsqu'il s'arrête
<code>docker run -d monimage</code>	Démarrer un conteneur en arrière-plan (en mode détaché)

### 2.1.2.2. Démarrer un conteneur Docker arrêté

Instruction	Description
<code>docker start monconteneur</code>	Redémarrer un conteneur arrêté
<code>docker start -ai monconteneur</code>	Redémarrer un conteneur en mode interactif (utile pour des commandes nécessitant des entrées dans la console, comme bash)

### 2.1.2.3. Exécuter des commandes dans un conteneur Docker en cours d'exécution

Instruction	Description
<code>docker exec monconteneur commande</code>	Exécuter une commande dans un conteneur en cours d'exécution
<code>docker exec -it monconteneur commande</code>	Exécuter une commande dans un conteneur en cours d'exécution et le remettre en mode interactif (utile pour des commandes nécessitant des entrées dans la console, comme bash)

### 2.1.2.4. Arrêter un conteneur en cours d'exécution

Instruction	Description
<code>docker stop monconteneur</code>	Arrêter un conteneur

### 2.1.2.5. Supprimer un conteneur Docker

Instruction	Description
<code>docker rm monconteneur</code> ou <code>docker container rm monconteneur</code>	Supprimer un conteneur

### 2.1.2.6. Afficher les conteneurs Docker

Instruction	Description
<code>docker ps</code> ou <code>docker container ls</code>	Afficher les différents conteneurs en cours d'exécution
<code>docker ps -a</code>	Afficher tous les conteneurs (même ceux arrêtés)

### 2.1.2.7. Autres commandes Docker

Instruction	Description
<code>docker attach monconteneur</code>	Remettre un conteneur au premier-plan
<code>docker cp monconteneur:/chemin... cheminHôte</code>	Récupérer des fichiers provenant d'un conteneur
<code>docker cp cheminHôte monconteneur:/chemin...</code>	Envoyer des fichiers dans un conteneur

### 2.1.3. Les volumes Docker

Instruction	Description
<code>docker volume create monVolume</code>	Créer un volume
<code>docker volume ls</code>	Afficher les volumes
<code>docker run -it --rm -v monVolume:dossierConteneur monimage</code>	Lier un dossier de l'hôte avec un dossier du conteneur
<code>docker volume rm monVolume</code>	Supprimer un volume
<code>docker volume inspect monVolume</code>	Afficher les détails du volume

### 2.1.4. Les partages de ports Docker

Instruction	Description
<code>docker network create --driver=bridge monreseau</code>	Créer un réseau partagé
<code>docker run --rm -it --network=none monimage</code>	Démarrer un conteneur isolé du réseau
<code>docker run --rm -p portLocal:portConteneur monimage</code>	Démarrer un conteneur avec un port partagé et accessible depuis l'hôte
<code>docker run -it --rm --network=monreseau monimage</code>	Démarrer un conteneur connecté à un réseau
<code>docker network connect monreseau monconteneur</code>	Connecter un conteneur à un réseau
<code>docker network disconnect monreseau monconteneur</code>	Déconnecter un conteneur du réseau
<code>docker network ls</code>	Afficher les réseaux disponibles
<code>docker network inspect monreseau</code>	Afficher les détails d'un réseau
<code>docker network rm monreseau</code>	Supprimer un réseau partagé

### 2.2. Le fichier dockerfile

Instruction	Description
<code>FROM image</code>	Récupérer les librairies d'une image existante (ex : ubuntu)
<code>RUN commande</code>	Exécuter une commande lors de la création de l'image (chaque RUN ajoute une couche au système de fichiers overlay)
<code>RUN commande1 &amp;&amp; commande2...</code>	Exécuter plusieurs commandes lors de la création de l'image dans une même couche
<code>COPY source destination</code>	Copier des fichiers ou des dossiers (chaque COPY ajoute une couche au système de fichiers overlay)
<code>ADD source destination</code>	Décompresser une archive ou télécharger depuis un lien internet et le copier dans une destination
<code>WORKDIR dossier</code>	Changer de répertoire
<code>EXPOSE port</code>	Changer le port à l'écoute (ex : 80)
<code>VOLUME dossier</code>	Partager un dossier avec le système de fichiers hôte
<code>ENTRYPOINT ["argument1", argument2"..."]</code>	Exécuter les commandes avec arguments sous forme de tableau ne pouvant être remplacées qu'avec la commande : <code>docker run --entrypoint="argument3</code>

	<i>argument4...</i> <i>monimage</i> (écrasant automatiquement les arguments du CMD)
ENTRYPOINT <i>argument1 argument2...</i>	Exécuter les commandes avec arguments ne pouvant être remplacées qu'avec la commande : <code>docker run --entrypoint="<i>argument3 argument4...</i>" <i>monimage</i></code> (écrasant automatiquement les arguments du CMD)
CMD [ <i>"argument1", argument2...</i> ]	Exécuter la commande avec des arguments sous forme de tableau si le conteneur n'a pas d'argument lors de l'exécution de la commande : <code>docker run <i>monimage argument3 argument4...</i></code> (sans écraser les arguments de l'ENTRYPOINT)
CMD <i>argument1 argument2...</i>	Exécuter la commande si le conteneur n'a pas d'argument lors de l'exécution de la commande : <code>docker run <i>monimage argument3 argument4...</i></code> (sans écraser les arguments de l'ENTRYPOINT)
ENV <i>nom valeur</i>	

## 3. Docker-Compose

### 3.1. Les commandes Docker-Compose de base

Les commandes ci-dessous nécessitent de privilèges administrateurs (avec `sudo` avant la commande).

Astuce : Pour ne pas avoir à utiliser tout le temps `sudo`, il est possible d'activer un compte root en définissant un mot de passe avec `sudo passwd root`, et d'ouvrir un terminal avec la commande : `su`

Remarque : si les commandes ci-dessous ne fonctionnent pas, essayez d'installer les autres packages nécessaires avec les commandes :

```
sudo apt install curl
```

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/download/v2.20.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

#### 3.1.1. Créer un conteneur Docker-Compose

Instruction	Description
<code>docker-compose up</code>	Démarrer un conteneur à partir d'un Docker-Compose (et tous les services)
<code>docker-compose up monconteneur</code>	Démarrer un conteneur à partir d'un Docker-Compose et choisir le nom du conteneur
<code>docker-compose up -d</code>	Démarrer un conteneur en arrière-plan

#### 3.1.2. Démarrer un conteneur Docker-Compose

Instruction	Description
<code>docker-compose start monconteneur</code>	Démarrer un conteneur déjà instancié



### 3.1.3. Arrêter un conteneur Docker-Compose

Instruction	Description
<code>docker-compose stop monconteneur</code>	Arrêter un conteneur en cours d'exécution

### 3.1.4. Supprimer un conteneur Docker-Compose

Instruction	Description
<code>docker-compose rm monconteneur</code>	Supprimer un conteneur

### 3.1.5. Autres commandes Docker-Compose

Instruction	Description
<code>docker-compose logs</code>	Afficher les traces du conteneur

## 3.2. Le fichier docker-compose.yml