Docker

Docker est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels lancée en 2013.

- Installation (debian)
- Mise à jour (debian)
- Purge

Installation (debian)

Mettre à jour votre système

Avant d'installer Docker, il est recommandé de mettre à jour votre système Debian pour s'assurer que tous les paquets sont à jour.

```
sudo apt update && sudo apt upgrade -y
```

Installer les dépendances nécessaires

Docker nécessite quelques outils comme `apt-transport-https`, `ca-certificates`, `curl`, `software-properties-common` et `gnupg2`. Installez-les en utilisant la commande suivante :

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg2 -y
```

Ajouter la clé GPG officielle de Docker

Docker signe ses paquets avec une clé GPG pour garantir leur authenticité. Vous devez ajouter cette clé pour pouvoir installer Docker depuis les dépôts officiels.

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
```

Ajouter le dépôt Docker

Ensuite, vous devez ajouter le dépôt officiel de Docker à votre liste de sources APT. Exécutez cette commande pour ajouter le dépôt stable :

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

Mettre à jour les sources

Une fois le dépôt Docker ajouté, vous devez mettre à jour la liste des paquets disponibles pour que le système reconnaisse les paquets Docker :

```
sudo apt update
```

Installer Docker

Vous pouvez maintenant installer Docker. Utilisez la commande suivante pour installer Docker CE (Community Edition) :

```
sudo apt install docker-ce docker-ce-cli containerd.io -y
```

Vérifier l'installation

Une fois Docker installé, vérifiez qu'il fonctionne correctement en exécutant la commande suivante pour voir la version de Docker installée :

```
docker --version
```

Démarrer et activer le service Docker

Par défaut, le service Docker devrait démarrer automatiquement après l'installation. Si ce n'est pas le cas, vous pouvez le démarrer manuellement et l'activer pour qu'il démarre automatiquement au démarrage de votre VM :

```
sudo systemctl start docker
sudo systemctl enable docker
```

Vérifier que Docker fonctionne correctement

Pour vérifier que Docker fonctionne correctement, exécutez la commande suivante, qui va télécharger et exécuter un conteneur test :

```
sudo docker run hello-world
```

Si tout est correctement installé, vous devriez voir un message vous indiquant que Docker fonctionne correctement.

Ajouter votre utilisateur au groupe Docker (facultatif)

Si vous souhaitez exécuter des commandes Docker sans avoir à utiliser `sudo` à chaque fois, vous pouvez ajouter votre utilisateur au groupe `docker` :

```
sudo usermod -aG docker $USER
```

Pour que ce changement prenne effet, vous devez vous déconnecter puis vous reconnecter à votre session ou exécuter la commande suivante :

```
newgrp docker
```

Mise à jour (debian)

Mettre à jour la liste des dépôts

La première étape consiste à mettre à jour la liste des dépôts pour s'assurer que vous avez les dernières informations sur les paquets disponibles.

```
sudo apt update
```

Vérifier les versions disponibles

Avant de mettre à jour Docker, vous pouvez vérifier si une nouvelle version est disponible. Pour cela, utilisez la commande suivante :

```
apt list --upgradable
```

Cela affichera la liste des paquets pouvant être mis à jour, y compris Docker si une nouvelle version est disponible.

Mettre à jour Docker

Si Docker est inclus dans la liste des paquets à mettre à jour, vous pouvez procéder à la mise à jour avec cette commande :

```
sudo apt upgrade docker-ce docker-ce-cli containerd.io
```

Cela mettra à jour Docker et ses composants principaux.

(Facultatif) Mettre à jour tous les paquets du système

Si vous voulez mettre à jour l'ensemble des paquets de votre système, y compris Docker et d'autres dépendances, utilisez la commande suivante :

```
sudo apt upgrade -y
```

Cela mettra à jour tous les paquets installés, y compris Docker, si une nouvelle version est disponible dans les dépôts.

Vérifier la version de Docker

Une fois la mise à jour terminée, vous pouvez vérifier que Docker a bien été mis à jour en exécutant :

```
docker --version
```

Cela vous affichera la version actuelle de Docker installée.

Redémarrer le service Docker (si nécessaire)

Bien que Docker mette généralement à jour ses composants sans redémarrage, il est parfois nécessaire de redémarrer le service Docker pour que les modifications prennent effet correctement :

```
sudo systemctl restart docker
```

Vérifier le fonctionnement de Docker

Pour vous assurer que Docker fonctionne correctement après la mise à jour, vous pouvez exécuter un conteneur de test :

```
sudo docker run hello-world
```

Si Docker fonctionne correctement, le message "Hello from Docker!" s'affichera.

Nettoyage (Facultatif)

Enfin, pour garder votre système propre, vous pouvez supprimer les anciennes versions de paquets qui ne sont plus nécessaires en exécutant :

```
sudo apt autoremove
```

Cela supprimera les dépendances inutiles et les anciens paquets qui ne sont plus utilisés.

Purge

Cette procédure permet de supprimer l'intégralité des données de Docker (conteneurs, images, volumes, réseaux et cache) pour retrouver un environnement "vierge", tout en conservant l'installation de Docker Engine.

---

1. Arrêt et suppression des conteneurs

La première étape consiste à stopper tous les processus en cours pour libérer les ressources.

- **Arrêter tous les conteneurs :**

```
docker stop $(docker ps -aq)
```

- **Supprimer tous les conteneurs :**

```
docker rm $(docker ps -aq)
```

---

2. Suppression des images

Pour libérer l'espace disque occupé par les systèmes de fichiers superposés.

- **Supprimer toutes les images (utilisées ou non) :**

```
docker rmi -f $(docker images -aq)
```

---

3. Suppression des volumes et réseaux

Les volumes contiennent vos données persistantes et les réseaux vos configurations de ponts personnalisées.

- **Supprimer tous les volumes :**

```
docker volume rm $(docker volume ls -q)
```

- **Supprimer les réseaux personnalisés :** *(Note : Les réseaux par défaut comme '*bridge*', '*host*' et '*none*' ne seront pas supprimés)*

```
docker network rm $(docker network ls -q)
```

---

4. Nettoyage du cache de construction (Build Kit)

Docker conserve des traces des étapes de construction pour accélérer les futurs `docker build`. Pour une purge totale, il faut les vider.

- **Supprimer le cache de build :**

```
docker builder prune -a -f
```

---

5. La commande "All-in-One" (Le Grand Ménage)

Si vous souhaitez exécuter l'intégralité de ces étapes en une seule ligne de commande sécurisée (elle vérifie d'abord s'il y a des éléments à supprimer pour éviter les erreurs de syntaxe), utilisez celle-ci :

```
docker stop $(docker ps -aq) 2>/dev/null; docker system prune -a --volumes -f
```

Détail de la commande All-in-One :

- `docker stop ...` : Arrête proprement tous les conteneurs.
- `2>/dev/null` : Masque les erreurs si aucun conteneur n'est en cours.
- `docker system prune` : La commande native de Docker pour le nettoyage.

- `-a` : Supprime toutes les images inutilisées, pas seulement les "dangling" (sans nom).
- `--volumes` : Force la suppression des volumes (non inclus par défaut dans le prune).
- `-f` : Force l'exécution sans demander de confirmation (mode automatique).

> **Attention :** Cette opération est irréversible. Toutes les données contenues dans les bases de données (volumes) et les configurations spécifiques seront perdues.