

Installation Docker Engine & Docker Compose

Prérequis

Avant de commencer, assurez-vous que votre système répond aux exigences suivantes :

1. **Système d'exploitation**: Une VM Debian.
2. **Droits d'administrateur**: Vous devez avoir accès à un compte utilisateur avec des droits `sudo` pour exécuter des commandes d'installation et de configuration.
3. **Connexion Internet**: Pour télécharger Docker et ses dépendances.
4. **Espace disque et mémoire**: Assurez vous d'avoir suffisamment d'espace disque et de mémoire pour Docker et les conteneurs que vous prévoyez d'exécuter.

Installation de Docker Engine sur Debian

Étape 1: Mise à jour du système

Commencez par mettre à jour votre liste de paquets et mettez à niveau les paquets existants pour vous assurer que votre système est à jour.

```
sudo apt-get update  
sudo apt-get upgrade -y
```

Étape 2: Installer les paquets nécessaires

Avant d'installer Docker Engine pour la première fois sur une nouvelle machine hôte, vous devez configurer le dépôt **apt de Docker**. Ensuite, vous pourrez installer et mettre à jour Docker à partir de ce dépôt.

```
# Add Docker's official GPG key:
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Étape 3: Installation de docker

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

Pour vérifier son fonctionnement :

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

Pour désinstaller docker :

```
sudo apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-ce-
rootless-extras
```

Si l'installation s'est bien déroulée, vous verrez un message indiquant que votre installation de Docker fonctionne correctement.

Post-installation (facultatif mais recommandé)

Configurer Docker pour l'utiliser sans sudo (facultatif)

Par défaut, l'exécution des commandes Docker nécessite des privilèges administratifs (sudo).

Pour éviter d'avoir à taper `sudo` chaque fois que vous utilisez la commande `docker`, ajoutez votre utilisateur au groupe Docker.

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

Après avoir exécuté cette commande, vous devrez vous déconnecter et vous reconnecter pour que les modifications prennent effet.

Démarrer Docker au démarrage

Pour vous assurer que Docker démarre automatiquement à chaque démarrage du système :

```
sudo systemctl enable docker
```

Source : [Docker Official](#)

Descriptif

Ces commandes, exécutées dans l'ordre, préparent un système Debian pour une installation sécurisée et vérifiée de Docker Engine directement depuis les sources officielles de Docker, assurant ainsi que vous obtenez la version la plus récente et authentique de Docker.

Ajout de la clé GPG officielle de Docker

`sudo apt-get update` : Cette commande met à jour l'index des paquets disponibles et leurs versions, mais sans installer ou mettre à jour aucun paquet. Cela assure que vous avez les dernières informations sur les paquets disponibles avant d'installer des dépendances.

`sudo apt-get install ca-certificates curl` : Installe `ca-certificates` et `curl` sur votre système.

`ca-certificates` : Ce paquet est nécessaire pour que votre système puisse vérifier les certificats SSL des sites web (ce qui est crucial pour des téléchargements sécurisés).

`curl` : Un outil en ligne de commande utilisé pour transférer des données avec des URL. Ici, il sert à télécharger la clé GPG de Docker depuis Internet.

`sudo install -m 0755 -d /etc/apt/keyrings` : Crée le répertoire `/etc/apt/keyrings` s'il n'existe pas déjà, avec des permissions telles que le propriétaire (root) peut lire, écrire et exécuter, tandis que les membres du groupe et les autres utilisateurs peuvent seulement lire et exécuter. Ce répertoire est utilisé pour stocker les clés GPG de confiance pour APT.

`sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc` : Télécharge la clé GPG officielle de Docker et la sauvegarde dans le fichier `/etc/apt/keyrings/docker.asc`. L'option `-fsSL` rend `curl` silencieux sauf en cas d'erreur, suit les redirections, et permet le protocole SSL.

`sudo chmod a+r /etc/apt/keyrings/docker.asc` : Modifie les permissions du fichier de clé GPG pour permettre à tous les utilisateurs de le lire. Cela est nécessaire pour que APT puisse vérifier les paquets signés par cette clé.

Ajout du dépôt Docker aux sources APT

La commande `echo` suivie de `sudo tee /etc/apt/sources.list.d/docker.list > /dev/null` : Cette commande complexe ajoute la ligne de dépôt officiel de Docker aux sources APT du système.

`echo` génère la ligne de texte qui définit le dépôt.

Le dépôt est configuré pour utiliser l'architecture système (`arch=$(dpkg --print-architecture)`), signé par la clé précédemment téléchargée (`signed-by=/etc/apt/keyrings/docker.asc`), pointe vers le dépôt Docker (`https://download.docker.com/linux/debian`), et spécifie la version du système d'exploitation (`$(. /etc/os-release && echo "$VERSION_CODENAME")`) et le canal `stable`.

`sudo tee /etc/apt/sources.list.d/docker.list > /dev/null` écrit la sortie de `echo` dans le fichier `/etc/apt/sources.list.d/docker.list` (créant ou écrasant le fichier si nécessaire), avec les droits d'administration. L'envoi de la sortie vers `/dev/null` empêche l'affichage de la ligne dans la console.

`sudo apt-get update` : Exécute de nouveau `apt-get update` pour que APT prenne en compte le nouveau dépôt ajouté. Après cette commande, le système est prêt à installer Docker Engine depuis le dépôt officiel ajouté.

Installation de Docker & Docker Compose

`docker-ce` : "Docker Community Edition" est le logiciel de base de Docker qui permet de créer, déployer et gérer des conteneurs. C'est le cœur de Docker, fournissant l'environnement d'exécution des conteneurs.

`docker-ce-cli` : "Docker Community Edition Command Line Interface" est l'interface en ligne de commande pour Docker. Elle permet aux utilisateurs d'interagir avec Docker et de gérer les conteneurs Docker à travers une variété de commandes.

`containerd.io` : Containerd est un environnement d'exécution de conteneurs disponible sous forme de daemon pour Linux et Windows. Il s'agit d'un composant de bas niveau pour la gestion du cycle

de vie des conteneurs, responsable de l'exécution des conteneurs, de la gestion de leur stockage d'images, de la mise en réseau, et plus encore. Docker utilise Containerd sous le capot.

[docker-buildx-plugin](#) : Docker Buildx est un plugin Docker qui fournit des fonctionnalités de build avec BuildKit, permettant la construction de l'image Docker avec un support amélioré pour la multi-architecture, le caching, et plus encore. Cela permet de créer des images Docker de manière plus efficace et avancée, y compris la possibilité de créer simultanément des images pour différentes architectures à partir d'un seul poste de commande.

[docker-compose-plugin](#) : Docker Compose est un outil pour définir et gérer des applications multi-conteneurs avec Docker. Le plugin [docker-compose-plugin](#) intègre Docker Compose directement dans l'écosystème Docker, permettant aux utilisateurs de définir et d'exécuter des applications multi-conteneurs à l'aide de fichiers YAML.

Cela simplifie le processus de configuration, d'exécution, et de gestion des applications conteneurisées en utilisant une syntaxe déclarative.

Revision #6

Created 10 March 2024 12:29:45 by MASSON Romain

Updated 11 March 2024 14:44:42 by MASSON Romain