

Utilisation de Docker Machine

Docker Machine est un outil très pratique pour créer, gérer et configurer des hôtes Docker, qu'ils soient locaux ou distants. Il facilite la gestion des machines virtuelles Docker, que ce soit sur des plateformes cloud comme AWS, Azure, ou Google Cloud, ou sur des environnements locaux comme VirtualBox ou VMware.

Voici comment gérer vos hôtes Docker à l'aide de **Docker Machine** et quelles sont les commandes les plus utiles pour l'administration des hôtes Docker.

1. Gérer des Hôtes Docker à l'Aide de Docker Machine

Étape 1 : Lister les machines Docker

Pour voir la liste de toutes les machines Docker gérées par Docker Machine, utilisez la commande `docker-machine ls`. Elle affiche des informations sur les hôtes Docker, y compris leur nom, leur état, leur adresse IP et le driver utilisé.

```
docker-machine ls
```

Exemple de sortie :

NAME	ACTIVE	DRIVER	STATE	URL
SWARM DOCKER	ERRORS			
my-docker-vm	*	virtualbox	Running	tcp://192.168.99.100:2376
			v19.03.8	
my-aws-docker-vm	-	amazonec2	Running	tcp://34.214.12.45:2376
			v19.03.8	

- **ACTIVE** : Indique si l'hôte est actuellement actif (un astérisque (*) signifie qu'il est actif).
- **STATE** : Indique l'état de la machine (par exemple, "Running" pour en fonctionnement).
- **URL** : L'URL à laquelle Docker Machine peut être contacté.
- **DOCKER** : La version de Docker installée sur la machine.

Étape 2 : Se connecter à un hôte Docker

Pour vous connecter à un hôte Docker via SSH, utilisez la commande `docker-machine ssh` suivie du nom de la machine.

```
docker-machine ssh my-docker-vm
```

Cela ouvre une session SSH sur la machine virtuelle Docker `my-docker-vm`, vous permettant de l'administrer directement.

Étape 3 : Accéder à Docker sur un hôte distant

Pour envoyer des commandes Docker vers un hôte spécifique, vous devez configurer votre environnement en utilisant la commande `docker-machine env`. Cette commande configure les variables d'environnement nécessaires pour interagir avec un hôte Docker distant.

```
eval $(docker-machine env my-docker-vm)
```

Cela définit les variables d'environnement pour que le client Docker fonctionne sur l'hôte `my-docker-vm`. Après avoir exécuté cette commande, toutes les commandes Docker (comme `docker ps`, `docker run`) seront envoyées à cet hôte distant.

Si vous voulez revenir à votre machine Docker locale, vous pouvez réinitialiser l'environnement avec :

```
eval $(docker-machine env -u)
```

Étape 4 : Démarrer et arrêter des hôtes Docker

Docker Machine vous permet de démarrer ou d'arrêter des machines virtuelles Docker à tout moment. Voici les commandes de base pour cela :

- **Démarrer une machine virtuelle :**

```
docker-machine start my-docker-vm
```

- **Arrêter une machine virtuelle :**

```
docker-machine stop my-docker-vm
```

- **Redémarrer une machine virtuelle :**

```
docker-machine restart my-docker-vm
```

Ces commandes sont utiles pour gérer l'état de vos machines, en particulier si vous avez un grand nombre de machines virtuelles Docker à gérer.

Étape 5 : Supprimer une machine virtuelle

Si vous n'avez plus besoin d'un hôte Docker, vous pouvez le supprimer en utilisant la commande `docker-machine rm`. Soyez prudent, car cela supprimera la machine virtuelle et toutes les données associées.

```
docker-machine rm my-docker-vm
```

Cette commande supprime la machine `my-docker-vm` de Docker Machine, mais elle n'affecte pas les machines déjà en fonctionnement. Si vous voulez supprimer une machine qui est en cours d'exécution, vous devez d'abord l'arrêter.

2. Commandes Utiles pour Docker Machine

Voici quelques-unes des commandes les plus courantes et utiles pour travailler avec **Docker Machine**.

Créer une machine virtuelle avec Docker Machine

Pour créer un nouvel hôte Docker sur une machine locale (VirtualBox) ou un fournisseur de cloud, utilisez la commande `docker-machine create`. Vous pouvez personnaliser cette commande avec différents **drivers** (comme `virtualbox`, `amazonec2`, `azure`, etc.) et des options pour définir les ressources.

Exemple pour créer un hôte Docker avec VirtualBox :

```
docker-machine create --driver virtualbox --virtualbox-cpu-count 2 --virtualbox-memory 4096 --virtualbox-disk-size 20000 my-docker-vm
```

Exemple pour créer un hôte Docker sur AWS :

```
docker-machine create --driver amazonec2 --amazonec2-region us-west-2 --amazonec2-instance-type t2.micro --amazonec2-access-key <AWS_ACCESS_KEY> --amazonec2-secret-key <AWS_SECRET_KEY> my-aws-docker-vm
```

Afficher les détails d'une machine virtuelle

Pour obtenir des informations détaillées sur une machine virtuelle, utilisez la commande `docker-machine inspect`. Cela vous donnera des détails sur les configurations de la machine, l'adresse IP, les paramètres de stockage, etc.

```
docker-machine inspect my-docker-vm
```

Obtenir l'adresse IP d'un hôte Docker

Si vous voulez connaître l'adresse IP de la machine virtuelle Docker, vous pouvez utiliser la commande `docker-machine ip`. Cela vous permet de connaître l'adresse à laquelle Docker est accessible.

```
docker-machine ip my-docker-vm
```

Obtenir les logs de la machine virtuelle

Si vous avez besoin de diagnostiquer des problèmes avec un hôte Docker, vous pouvez afficher les logs de la machine avec la commande `docker-machine logs` :

```
docker-machine logs my-docker-vm
```

Mettre à jour Docker Machine

Pour mettre à jour Docker Machine vers la dernière version disponible, vous pouvez utiliser les commandes de mise à jour propre à chaque plateforme. Par exemple, sur Linux, vous pouvez réinstaller le binaire avec `curl` :

```
curl -L  
https://github.com/docker/machine/releases/download/v0.16.2/docker-  
machine-$(uname -s)-$(uname -m) > /usr/local/bin/docker-machine  
sudo chmod +x /usr/local/bin/docker-machine
```

Sur macOS, vous pouvez utiliser **Homebrew** :

```
brew upgrade docker-machine
```

Sur Windows, vous pouvez utiliser **Chocolatey** :

```
choco upgrade docker-machine
```

Autres Commandes Docker Machine

- **Afficher l'environnement de la machine Docker** (pour configurer votre shell pour interagir avec une machine spécifique) :

```
docker-machine env my-docker-vm
```

- **Mettre à jour le client Docker avec une version spécifique :**

```
docker-machine upgrade my-docker-vm
```

Résumé des Commandes Docker Machine

- **docker-machine create** : Crée un hôte Docker sur la machine locale ou cloud.
- **docker-machine ls** : Affiche la liste des hôtes Docker existants.
- **docker-machine ssh** : Se connecte à un hôte Docker via SSH.
- **docker-machine env** : Configure l'environnement du client Docker pour interagir avec un hôte distant.
- **docker-machine start / docker-machine stop / docker-machine restart** : Démarre, arrête ou redémarre un hôte Docker.
- **docker-machine rm** : Supprime un hôte Docker.
- **docker-machine inspect** : Affiche les détails d'un hôte Docker.
- **docker-machine ip** : Affiche l'adresse IP d'un hôte Docker.
- **docker-machine logs** : Affiche les logs d'un hôte Docker.

Docker Machine simplifie la gestion de vos hôtes Docker en offrant une interface de commande unique pour créer, gérer et interagir avec les machines Docker, que ce soit localement ou dans le cloud.