

Configurer le démon Dockerd

Le démon Docker (**dockerd**) est le cœur du système Docker. Il gère les conteneurs, les images, les réseaux et les volumes, et communique avec le client Docker (**docker**) pour exécuter des commandes. Configurer **dockerd** permet de personnaliser son comportement, comme l'allocation des ressources, la gestion des réseaux et des volumes, ainsi que la sécurité du système. Voici une vue d'ensemble de la configuration du démon Docker.

Configuration des paramètres du démon

1. Démarrage automatique du démon Docker :

Le démon Docker démarre automatiquement lorsque le système est lancé, mais il peut être configuré pour démarrer avec des options spécifiques.

- **Système de gestion des services** : Docker peut être géré par un gestionnaire de services comme **systemd** ou **upstart**. Si vous utilisez **systemd**, vous pouvez configurer des paramètres en modifiant le fichier d'unité Docker.

Exemple pour configurer **dockerd** avec **systemd** :

```
sudo systemctl edit docker.service
```

Vous pouvez alors ajouter des options de configuration dans cette section.

2. Paramètres de ressources :

Le démon Docker peut être configuré pour limiter les ressources disponibles (CPU, mémoire, etc.) ou pour changer la façon dont il gère ces ressources. Par exemple :

- **--default-ulimit** : Définit les limites par défaut de ressources pour les conteneurs.
- **--max-concurrent-downloads** : Limite le nombre de téléchargements simultanés d'images Docker.

3. Paramètres de logging :

Docker prend en charge plusieurs pilotes de journaux (logs). Vous pouvez définir le pilote de journaux à utiliser pour le démon Docker, ce qui peut être utile pour centraliser les journaux dans un système de gestion de logs comme ELK (Elasticsearch, Logstash, Kibana).

Exemple d'option de logging :

```
--log-driver=syslog
```

4. Gestion du stockage :

Vous pouvez configurer le système de stockage Docker avec l'option **--storage-driver**, en fonction des besoins de performance. Par exemple, pour utiliser le stockage **overlay2** (recommandé), vous pouvez configurer :

```
--storage-driver=overlay2
```

Fichiers de configuration

1. Fichier de configuration principal (/etc/docker/daemon.json) :

Docker permet de configurer de nombreux paramètres à l'aide d'un fichier JSON. Ce fichier est situé par défaut à `/etc/docker/daemon.json` (sur Linux), et il contient les configurations par défaut du démon Docker.

Exemple d'un fichier `daemon.json` :

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "10m",
    "max-file": "3"
  },
  "storage-driver": "overlay2",
  "default-runtime": "runc",
  "insecure-registries": ["myregistry.local:5000"],
  "live-restore": true
}
```

Explication des options :

- o `"log-driver"` : Spécifie le pilote de journalisation utilisé pour les conteneurs Docker.
- o `"log-opts"` : Paramètres de configuration du journal, comme la taille maximale des fichiers journaux et leur nombre.
- o `"storage-driver"` : Définit le mécanisme de stockage des images et des conteneurs (ex. `overlay2`, `aufs`).
- o `"insecure-registries"` : Liste des registres Docker non sécurisés (pour les tests, par exemple).
- o `"live-restore"` : Permet de maintenir Docker fonctionnel pendant la mise à jour du démon.

2. Autres fichiers de configuration :

- o `/etc/default/docker (Ubuntu/Debian)` : Ce fichier permet de définir des options supplémentaires au démarrage du démon Docker via `systemd` ou autre gestionnaire de services. Par exemple :

```
DOCKER_OPTS="--host tcp://0.0.0.0:2375 --insecure-registry  
registry.example.com"
```

Options de démarrage

Lorsque vous lancez le démon Docker (**dockerd**), vous pouvez ajouter diverses options pour modifier son comportement au démarrage.

1. Exemple de lancement avec options :

Pour démarrer **dockerd** avec des options personnalisées, vous pouvez lancer directement la commande suivante :

```
sudo dockerd --host tcp://0.0.0.0:2375 --log-driver=syslog --storage-driver=overlay2
```

2. Options courantes de démarrage :

- **--host** : Spécifie l'adresse à laquelle le démon Docker doit écouter. Par exemple, pour permettre l'accès à Docker à distance via un réseau, vous pouvez lier Docker à une adresse IP :

```
--host tcp://0.0.0.0:2375
```

- **--log-driver** : Détermine le moteur de journalisation (par exemple, **json-file**, **syslog**).
- **--storage-driver** : Choisir le type de driver de stockage des images (par exemple, **overlay2**, **aufs**).
- **--insecure-registries** : Permet de définir des registres non sécurisés :

```
--insecure-registries registry.example.com
```

- **--live-restore** : Active la fonctionnalité permettant à Docker de continuer à fonctionner même si le démon est redémarré, ce qui aide lors des mises à jour ou redémarrages planifiés.

3. Démarrage avec **systemd** :

Sur les systèmes utilisant **systemd**, la configuration du démon Docker est généralement gérée via un fichier d'unité. Vous pouvez ajouter des options supplémentaires dans le fichier d'unité de Docker (**/etc/systemd/system/docker.service**).

Exemple d'ajout d'options dans le fichier d'unité :

```
[Service]
ExecStart=/usr/bin/dockerd --host tcp://0.0.0.0:2375
```

Ensuite, pour recharger la configuration et appliquer les changements :

```
sudo systemctl daemon-reload  
sudo systemctl restart docker
```

Conclusion

La configuration du démon Docker (**dockerd**) est essentielle pour optimiser le fonctionnement de Docker selon les besoins spécifiques d'une organisation ou d'un projet. En configurant les paramètres de démarrage, les ressources, le stockage et les logs, vous pouvez personnaliser Docker pour mieux s'adapter à votre environnement de travail. Le fichier **daemon.json** et les options passées directement au démarrage sont les deux moyens principaux de configurer **dockerd**.