
docker-daemon

Activer le mode service

OPTIONS

- api-cors-header=""** Définis les en-têtes CORS dans l'API distante. Défaut : désactivé.
- authorization-plugin=""** Définis les plugins d'autorisation à charger
- b, -bridge=""** Attache les conteneurs à un bridge réseau existant
- bit=""** Utiliser l'adresse en notation CIDR donnée pour la création du bridge0. Ne peut pas être utilisé avec **-b**
- cgroup-parent=""** Définis le cgroup parent pour tous les conteneurs. Défaut : /docker pour le pilote cgroup fs, et system.slice pour le pilote cgroup systemd
- cluster-store=""** URL du backend de stockage distribué
- cluster-advertise=""** Spécifie la combinaison 'host :port' ou 'interface :port' que cette instance particulière devrait utiliser en s'annonçant au cluster. Le service est accessible via cette valeur.
- cluster-store-opt=""** Spécifie les options pour la magasin de clé/valeurs
- config-file="/etc/docker/daemon.json"** Spécifie le fichier de configuration JSON à charger
- D, -debug=truelfalse** Active le mode debug. Défaut : false
- default-gateway=""** Adresse IPv4 de la passerelle par défaut du conteneur. Cette adresse doit faire partie du bridge, définis par **-bip**
- default-gateway-v6=""** idem avec l'IPv6
- default-ulimit= []** Définis les ulimits pour les conteneurs
- disable-legacy-registry=truelfalse** Ne pas contacter les registre legacy
- dns=""** Forcer Docker à utiliser les serveurs DNS spécifiés
- dns-opt=""** Options DNS à utiliser
- dns-search= []** Domaines DNS à utiliser
- exec-opt= []** Définis les options du pilote d'exécution.
- exe-root=""** Chemin racine du pilote d'exécution Docker. Défaut : /var/run/docker
- fixed-cidr=""** Sous-réseau IPv4. Ce sous-réseau doit être imbriqué dans le bridge définis par **-bip**
- fixed-cidr-v6=""** Idem en ipv6
- G, -group=""** Groupe pour assigner un socker unix spécifié pas **-H**. Défaut : docker
- g, -graph=""** Chemin à utiliser comme racine du Docker runtime. Défaut : /var/lib/docker
- H, -host= [unix ://var/run/docker.sock] tcp ://[host :port] à connecter ou unix ://[path/to/socket] ou fd ://socketfd à utiliser. Peut être spécifiés plusieurs fois**
- icc=truelfalse** Autorise les communication inter-conteneur et l'hôte non-restreint. Si désactivé, les conteneurs peuvent être liés ensemble avec l'option **-link**. Défaut : true
- insecure-registry= []** Active la communication au registre non-sécuré.
- ip=""** Adresse IP par défaut à utiliser en se connectant aux ports du conteneur. Défaut : 0.0.0.0
- ip-forward=truelfalse** Active l'IP forwarding dans l'hôte Docker. Défaut : true. Intéragit avec l'ip forwarding dans le kernel.
- ip-masq=truelfalse** Active l'IP masquerading pour la plage IP du bridge. Défaut : true
- iptables=truelfalse** Active l'ajout de règles iptables par docker. Défaut : true
- ipv6=truelfalse** Active le support d'IPv6. Défaut : false. Docker créé un bridge avec l'adresse fe80 : :1.
- l, -log-level="debug|info|warn|error|fatal"** Niveau de log. Défaut : info

dm.blkdiscard Active l'utilisation de blkdiscard en supprimant les périphériques devicemapper. Désactivé par défaut à cause d'une latence additionnelle.

dm.override_udev_sync_check Par défaut, le backend devicemapper tente de se synchroniser avec udev pour le kernel linux. Cette option permet de désactiver cette synchronisation.

Options de stockage en cluster

Le service utiliser libkv pour annoncer le nœud dans le cluster. Certains backend key/value supportent TLS, et les paramètres TLS client utilisées par le service peuvent être configurés avec `–cluster-store-opt` :

kv.cacertfile Certificats d'autorité au format PEM

kv.certfile Certificat PEM à utiliser

kv.keyfile Clé privée PEM à utiliser

Autorisation d'accès

L'autorisation d'accès à Docker peut être étendue par des plugins d'autorisation. Il est possible d'utiliser plusieurs plugins d'autorisation en même temps, avec l'option `–authorization`.