
docker-create

Créer un nouveau conteneur

Créer un nouveau conteneur basé sur l'image spécifiée et la prépare pour exécuter la commande spécifiée. L'ID du conteneur est affiché sur STDOUT. C'est similaire à `docker run` excepté que le conteneur n'est jamais démarré. Le status initial du conteneur créé avec `docker create` est 'created'

OPTIONS

- a, --attach= []** Attache à STDIN, STDOUT ou STDERR. En mode foreground.
- add-host= []** Ajoute un mappage host-to-IP personnalisé (host :ip). Ajoute une ligne à `/etc/hosts`. peut être spécifié plusieurs fois.
- blkio-weight=0** Poid IO entre 10 et 1000
- blkio-weight-device= []** Poid IO au format `DEVICE_NAME :WEIGHT`
- cpu-shares=0** Partage CPU
- cap-add= []** Ajouter des capabilities Linux
- cap-drop= []** Supprimer de capabilities Linux
- cgroup-parent=""** Chemin des cgroups sous lesquels créer le conteneur. Si le chemin n'est pas absolu, le chemin est relatif aux cgroups du processus init.
- cidfile=""** Écrit l'ID de conteneur dans le fichier
- cpu-period=0** Limite la période CFS
- cpuset-cpus=CPUSET-CPUS** CPU dans lesquels autoriser l'exécution
- cpuset-mems=""** Nœuds mémoire autorisés.
- cpu-quota=0** Limite le quota CPU CFS
- d, --detach=true|false** Lance de conteneur en tâche de fond et affiche l'ID du conteneur. défaut : false
- detach-keys=""** Change la séquence de touche pour détacher le conteneur. Défaut : CTRL-p CTRL-q.
- device= []** Un périphérique hôte pour le conteneur (ex `--device=/dev/sdc :/dev/xvdc :run`)
- device-read-bps= []** Limite le taux de lecture d'un périphérique (ex `--device-read-bps=/dev/sda :1mb`)
- device-read-iops** Limite le taux de lecture d'un périphérique (ex `--device-read-iops=/dev/sda :1000`)
- device-write-bps= []** Limite le taux d'écriture d'un périphérique (ex `--device-write-bps=/dev/sda :1mb`)
- device-write-iops** Limite le taux d'écriture d'un périphérique (ex `--device-write-iops=/dev/sda :1000`)
- dns-search= []** Définis les domaines de recherche DNS.
- dns-opt= []** Définis les options DNS
- dns= []** Définis les serveurs DNS
- e, --env= []** Définis des variables d'environnement.
- entrypoint=""** Remplace l'ENTRYPOINT de l'image
- env-file= []** Line un fichier de variables d'environnements
- expose= []** Expose un port, une une plage de ports informant Docker que le conteneur écoute sur les ports spécifiés. Docker utilise cette information pour interconnecter les conteneurs en utilisant des liens et pour définir la redirection de port dans le système hôte.
- group-add= []** Ajoute des groupes additionnels à lancer
- h, --hostname=""** nom d'hôte du conteneur
- i, --interactive=true|false** Garde STDIN ouvert même s'il n'est pas attaché. défaut : false
- ip=""** Définis l'adresse IPv4 de l'interface du conteneur, uniquement avec `--net` pour les réseaux utilisateurs

-ip6="" Définis l'adresse IPv6 de l'interface du conteneur, uniquement avec **-net** pour les réseaux utilisateurs

-ipc="" Créé un espace de nom IPC privé pour le conteneur

-isolation="default" spécifique le type de technologie d'isolation utilisée par les conteneurs

-l, -label= [] Définis les métadonnées du conteneur (ex : **-label com.example.key=value**)

-kernel-memory="" Limite la mémoire kernel. (0 = illimité)

-label-file= [] Lit un fichier de labels

-link= [] Ajoute un lien vers un autre conteneur sous la forme **<nom ou id> :alias**.

-log-driver="json-file|syslog|journald|gelf|fluentd|awslogs|splunk|none" Pilote de logging pour le conteneur. La commande **docker logs** ne fonctionne que pour les pilotes **json-file** et **journald**.

-log-opt= [] options spécifique au pilote de logging

-m, -memory="" Limite mémoire

-memory-reservation="" Limite soft

-memory-swap="LIMIT" Une valeur limite égale à la mémoire plus le swap. Doit être utilisé avec **-m**.

-mac-address="" L'adresse mac de conteneur

-name="" Assigne un nom au conteneur au format UUID long, UUID court, ou un nom.

-net="bridge" Définis le mode réseau pour le conteneur (**bridge**, **none**, **container :<namelid>**, **host**, **<network-name>|<network-id>**)

-net-alias= [] ajoute un alias réseau pour le conteneur

-oom-kill-disable=true|false Active ou non le OOM Killer pour le conteneur

-oom-score-adj="" Ajustement OOM de l'hôte pour le conteneur (-1000 à 1000)

-P, -publish-all=true|false Publie tous les ports exposés aux ports aléatoire dans les interfaces de l'hôte. Défaut : **false**

-p, -publish= [] Publie le port d'un conteneur, ou une plage de port, à l'hôte (Format : **ip :hostPort :containerPort | ip : :containerPort | hostPort :containerPort | containerPort**)

-pid=host Définis de mode PID pour le conteneur (**host** : utilise l'espace de nom PID de l'hôte, non sécurisé)

-uts=host Définis le mode UTS pour le conteneur (**host** : utilise l'espace de noms UTS dans le conteneur, non sécurisé)

-privileged=true|false Donne des privilèges étendus à ce conteneur. Défaut : **false**

-read-only=true|false Monte le système de fichier racine du conteneur en lecture seule.

-restart="no" Stratégie de redémarrage quand le conteneur se termine. (**no**, **on-failure [:max-retry]**, **always**, **unless-stopped**)

-rm=true|false Supprime automatiquement le conteneur quand il se termine. incompatible avec **-d**. défaut : **false**

-security-opt= [] Options de sécurité

-stop-signal=SIGTERM Signal pour stopper un conteneur

-shm-size="" Taille de **/dev/shm**. format : **<number><unit>**

-t, -tty=true|false Alloue un pseudo-TTY. défaut : **false**

-tmpfs= [] Créé un montage **tmpfs**

-u, -user="" Définis le username ou l'UID utilisé et optionnellement le group ou GID pour la commande spécifiée

-ulimit= [] Options **ulimit**

-v|-volume [= [[HOST-DIR :] CONTAINER-DIR [:OPTIONS]]] Créé un montage. avec **-v /HOST-DIR :/CONTAINER-DIR**, Docker monte **/HOST-DIR** dans l'hôte dans **/CONTAINER-DIR** dans le conteneur. Si **HOST-DIR** est omis, Docker créé un nouveau volume dans l'hôte. par défaut, les montages sont privés (non-visibles dans l'hôte). Peut être spécifié plusieurs fois.

-volume-driver="" Pilote de volume du conteneur. Ce pilote créé les volumes spécifié soit avec l'instruction **VOLUME** du **dockerfile**, ou depuis l'option **-v**.

-volumes-from= [] Monte les volumes depuis les conteneurs spécifiés.

-w, -workdir="" Répertoire de travail dans le conteneur