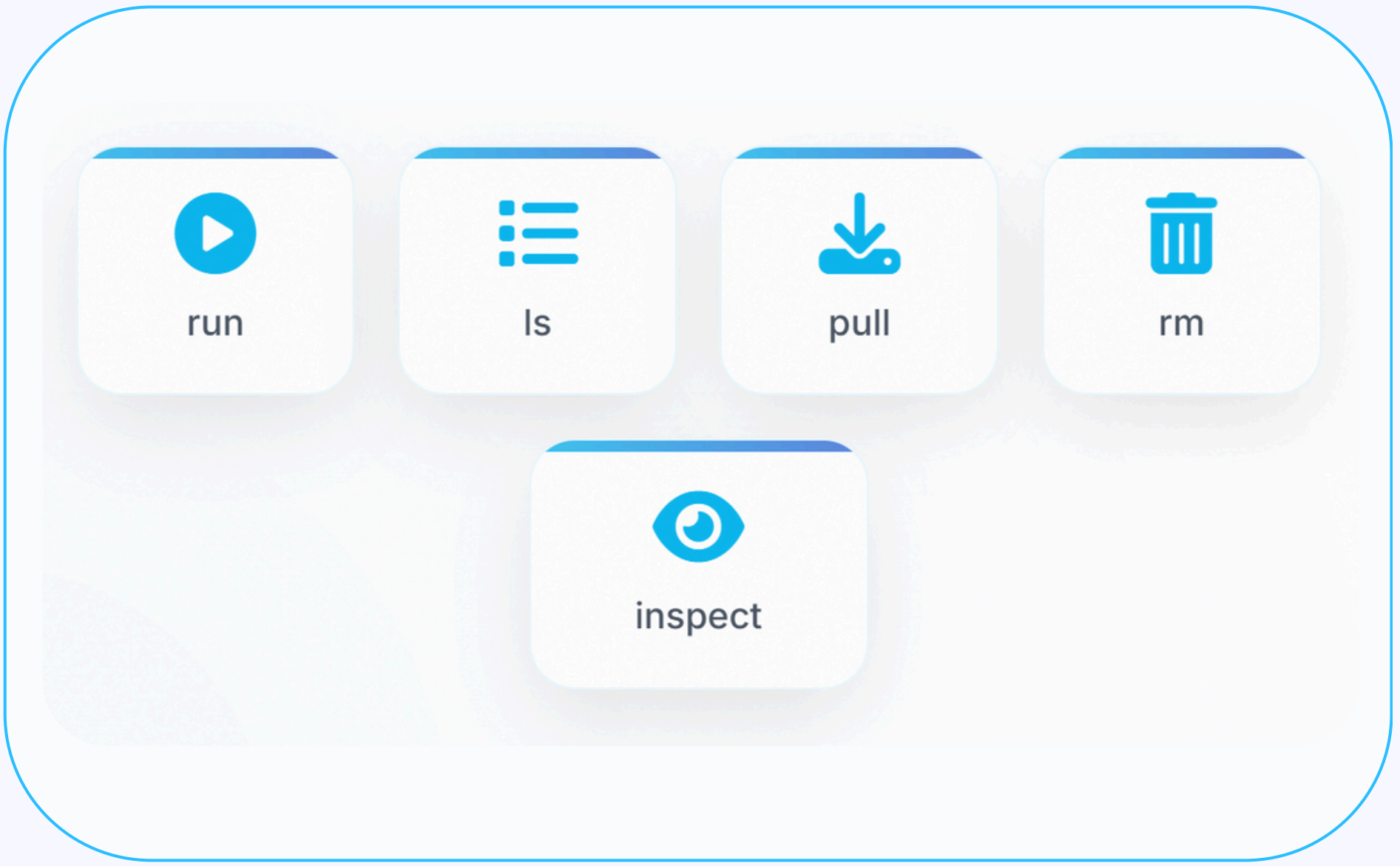
Yasser Kamal

#DevOps #CLI

Docker CLI

Les commandes Docker CLI essentielles pour démarrer, gérer et inspecter vos environnements conteneurisés.



La différence expliquée

À retenir

Une **image** est un plan fixe qu'on ne peut pas modifier. Un **conteneur** est une version "vivante" de l'image, qui peut changer et qu'on peut démarrer, arrêter ou supprimer.

En programmation

Si une image est une **classe**, un conteneur est une **instance** de cette classe. Vous pouvez créer plusieurs conteneurs à partir de la même image, comme vous pouvez créer plusieurs objets à partir de la même classe.

Obtenir des images

Les images Docker sont comme des applications qu'on télécharge d'un magasin d'apps “Docker Hub”.

Télécharger une image

```
$ docker image pull ubuntu
   Using default tag: latest
   latest: Pulling from library/ubuntu
   345e3491a907: Pull complete
   Digest: sha256:...
   Status: Downloaded newer image for ubuntu:latest
```

Bon à savoir

Lorsque vous n'indiquez pas de version avec : (comme **ubuntu:22.04**), Docker utilise automatiquement la version **latest**.

Voir vos images

```
$ docker image ls
   REPOSITORY    TAG     IMAGE         ID    CREATED       SIZE
   ubuntu        latest  825d55fb6340  2     weeks ago     72.8MB
   hello-world   latest  feb5d9fea6a5  5     months ago    13.3kB
```

Lancer des conteneurs

Pour créer et démarrer un conteneur à partir d'une image, on utilise la commande docker container run.

Lancer un conteneur

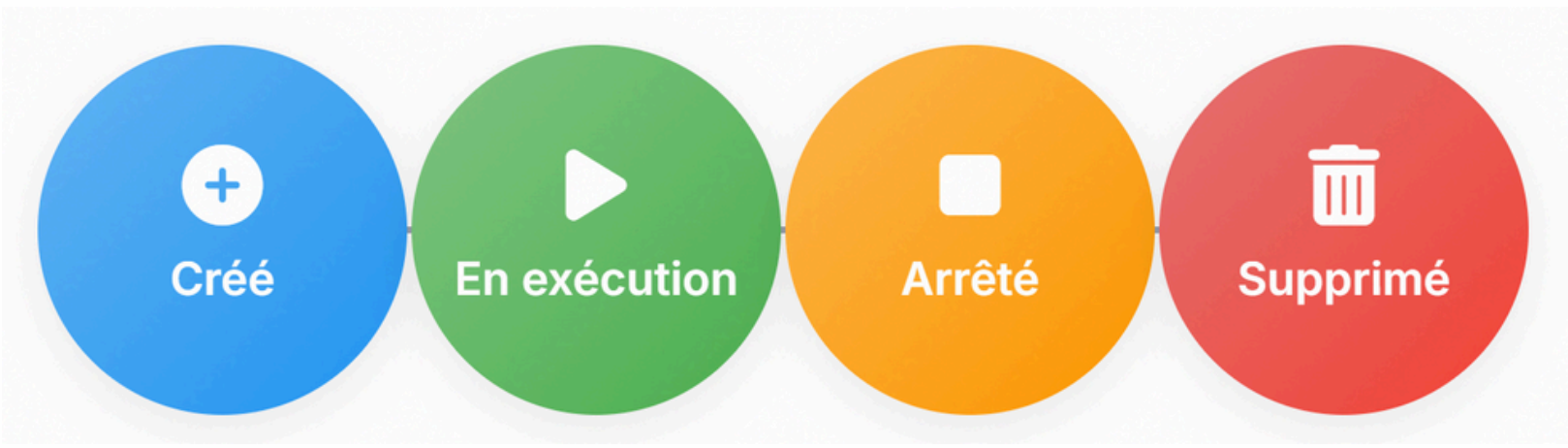
```
$ docker container run hello-world
   Hello from Docker!
   This message shows that your installation
   appears to be working correctly.
   ...
```

Ce qui se passe vraiment

Quand vous exécutez cette commande :

1. Docker cherche l'image sur votre ordinateur
2. S'il ne la trouve pas, il la télécharge
3. Docker crée un conteneur à partir de cette image
4. Docker exécute la commande par défaut de l'image
5. Le conteneur s'arrête quand la commande est terminée

Lancer des conteneurs



À savoir

run lance un nouveau conteneur à partir d'une image.
stop interrompt un conteneur en cours d'exécution.
start relance un conteneur déjà créé et arrêté.
rm supprime un conteneur arrêté du système.
⚠️ Un conteneur arrêté reste stocké tant que **rm** n'est pas exécuté.

Les 4 commandes clés

```
$ docker container run ubuntu
$ docker container stop mon_conteneur
$ docker container start mon_conteneur
$ docker container rm mon_conteneur
```

Voir vos conteneurs

Lister les conteneurs

```
$ docker container ls
   CONTAINER ID   IMAGE      COMMAND     CREATED        STATUS          NAMES

$ docker container ls -- all
   CONTAINER ID   IMAGE       COMMAND     CREATED        STATUS          NAMES
   f8b545a65ad9   hello-world  "/hello"   2 minutes      Exited (0)2     keen_wilson
                                          ago            minutes ago
```

Astuce

docker container ls montre seulement les conteneurs en cours d'exécution. Pour voir tous les conteneurs, même ceux qui sont arrêtés, utilisez docker container ls --all (ou -a en version courte).

Options pratiques

Les options (ou "flags") sont essentielles pour personnaliser le comportement de vos conteneurs. Pensez-y comme aux réglages d'un appareil : ils déterminent comment votre conteneur fonctionnera.

Mode Interactif (-it)

Combine -i (interactif) et -t (terminal) pour obtenir un shell interactif dans le conteneur.

```
$ docker run -it ubuntu bashubuntu
```

Idéal pour : Déboguer, explorer un conteneur

Nommer (--name)

Assigne un nom personnalisé au conteneur au lieu d'un nom aléatoire généré par Docker.

```
$ docker run --name ma_db mysql
```

Avantage : Référencer facilement un conteneur par son nom

Options pratiques

Auto-suppression (--rm)

Supprime automatiquement le conteneur dès qu'il s'arrête pour éviter l'accumulation.

```
$ docker run --rm alpine ls
```

Idéal pour : Tests rapides, opérations ponctuelles

Mode détaché (-d)

Lance le conteneur en arrière-plan, sans bloquer le terminal (service).

```
$ docker run -d nginx
```

Pour les logs : docker logs [CONTAINER]

Variables d'env. (-e)

Définit des variables d'environnement dans le conteneur pour le configurer.

```
$ docker run -e DB_HOST=mongodb mon_app
```

Conseil : Utilisez plusieurs -e pour définir plusieurs variables

Options pratiques

Mapping de ports (-p)

Connecte un port de la machine hôte à un port du conteneur pour y accéder.

```
$ docker run -p 8080:80 nginx
```

Format : -p HÔTE:CONTENEUR

Combinaison d'options : Mise en pratique

Exemple complet

```
$ docker container run -d --name mon_site -p 8080:80 -e
"LANG=fr_FR" --rm nginx
   Cette commande :
     1. Lance un serveur Nginx en arrière-plan (-d)
     2. Le nomme "mon_site" (--name)
     3. Rend le site accessible sur localhost:8080 (-p)
     4. Configure la langue en français (-e)
     5. Supprimera automatiquement le conteneur à son arrêt (--
rm)
```

Nettoyer votre système

Les conteneurs et images s'accumulent avec le temps et peuvent prendre beaucoup d'espace disque. Voici comment faire le ménage :

Nettoyage simplifié

```
$ docker container prune
   WARNING! This will remove all stopped containers.
   Are you sure you want to continue? [y/N] y
   Deleted Containers: f8d3b2a1...
```

En termes simples

La commande prune est comme un aspirateur qui nettoie tous les conteneurs arrêtés d'un coup. C'est rapide et efficace pour libérer de l'espace.

Nettoyer votre système

Astuce

Pour éviter d'avoir à nettoyer manuellement, utilisez toujours l'option --rm lorsque vous lancez des conteneurs temporaires :

Nettoyage --rm

```
$ docker container run --rm alpine echo "Je me supprime automatiquement!"
```

Grand nettoyage

```
$ docker system prune
   WARNING! This will remove:
     - all stopped containers
     - all networks not used by at least one container
     - all dangling images
     - all dangling build cache
   Are you sure you want to continue? [y/N] y
```

Yasser Kamal

#DevOps #Cloud

“Pour approfondir certains concepts abordés ici mais non développés en détail, n'hésitez pas à consulter les publications précédentes. Elles vous apporteront des éclairages complémentaires.”