

? Docker Cheat Sheet

Cette page regroupe les commandes Docker les plus utilisées sur la plateforme Homelab.

Elle s'applique principalement aux services déployés dans des conteneurs Docker sur des LXC ou des VM (Bitwarden, Harbor, Docker Registry, Code-Server, etc.).

? Objectif

Cette documentation permet de :

- Contrôler les conteneurs Docker
 - Administrer les stacks Docker Compose
 - Consulter les logs
 - Construire des images
 - Pousser des images vers le registre privé
 - Effectuer les opérations de maintenance courantes
-

? Vérifier l'état des conteneurs

Afficher les conteneurs en cours d'exécution

```
docker ps
```

Afficher tous les conteneurs

```
docker ps -a
```

Afficher les ressources utilisées

```
docker stats --no-stream
```

? Consulter les logs

Afficher les 50 dernières lignes

```
docker logs --tail 50 <container>
```

Suivre les logs en temps réel

```
docker logs -f --tail 50 <container>
```

?? Administrer un conteneur

Entrer dans un conteneur (Bash)

```
docker exec -it <container> /bin/bash
```

Si Bash n'est pas disponible (Alpine)

```
docker exec -it <container> /bin/sh
```

Redémarrer un conteneur

```
docker restart <container>
```

Arrêter un conteneur

```
docker stop <container>
```

Démarrer un conteneur

```
docker start <container>
```

? Docker Compose

La majorité des services de la plateforme sont déployés via **Docker Compose**.

Démarrer la stack

```
docker compose up -d
```

Arrêter la stack

```
docker compose down
```

Redémarrer la stack

```
docker compose restart
```

Mettre à jour les images

```
docker compose pull  
docker compose up -d
```

Docker Compose ne recrée que les services dont l'image a changé.

Consulter les logs

```
docker compose logs -f --tail 50
```

? Maintenance Docker

Afficher l'espace disque utilisé

```
docker system df
```

Nettoyer les ressources inutilisées

```
docker system prune
```

Nettoyage complet

```
docker system prune -a --volumes
```

⚠ **Attention** : cette commande supprime toutes les images inutilisées, les volumes orphelins et les conteneurs arrêtés.

? Construire une image

Depuis le dossier contenant le `Dockerfile`.

```
docker build -t registry.numericare.fr/library/code-server:custom-v1 .
```

? Registre privé

Connexion

```
docker login registry.numericare.fr
```

Pousser une image

```
docker push registry.numericare.fr/library/code-server:custom-v1
```

? Réseau

Afficher les réseaux Docker

```
docker network ls
```

Inspecter le réseau bridge

```
docker network inspect bridge
```

? Volumes

Afficher les volumes

```
docker volume ls
```

Inspecter un volume

```
docker volume inspect <volume>
```

? Dépannage

Problème	Commande
Le conteneur ne démarre pas	<code>docker logs <container></code>
Vérifier les ressources	<code>docker stats</code>
Problème réseau	<code>docker network inspect bridge</code>
Problème de volume	<code>docker volume inspect <volume></code>
Espace disque saturé	<code>docker system df</code>

? Bonnes pratiques

- Utiliser **Docker Compose** plutôt que des commandes `docker run`.
- Éviter les conteneurs créés manuellement.
- Toujours versionner les fichiers `docker-compose.yml`.
- Privilégier le registre privé **registry.numericare.fr**.

- Ne jamais modifier un conteneur en production avec `docker exec` sauf pour du diagnostic.
 - Reconstruire l'image plutôt que modifier un conteneur existant.
 - Nettoyer régulièrement les images inutilisées afin d'éviter la saturation des petits LXC.
-

? Voir également

- Installation de Docker sur un LXC Proxmox
- Docker Registry
- Harbor
- Code-Server
- Docker Compose Standards

...

Revision #1

Created 2026-06-26 09:41:05 UTC by Nico là

Updated 2026-06-26 09:41:32 UTC by Nico là