

# Cheat sheet Docker : toutes les commandes essentielles en un seul endroit

Résumez avec:



Share:



Docker est une plateforme open-source populaire qui facilite la création, le test, le déploiement et la gestion d'applications conteneurisées dans un environnement

## Nous nous soucions de votre vie privée

Nous utilisons des cookies, pour nous assurer que nous vous offrons la meilleure expérience sur notre site internet, pour permettre des services et des fonctionnalités essentiels et pour collecter des données sur la façon dont les visiteurs interagissent avec notre site et nos services. En cliquant sur Accepter, vous acceptez que nous utilisions tous les cookies pour la publicité, l'analyse et le support. Pour plus d'informations, veuillez lire notre [Politique de cookies](#).

Refuser

Paramètres

Accepter

[Commandes de service >](#)

[Commandes de réseau >](#)

## Architecture Docker

L'architecture Docker se compose de cinq éléments principaux : le serveur, le client, le conteneur, l'image et le registre.

### Serveur Docker

Un serveur Docker ou démon Docker est un programme qui s'exécute en arrière-plan sur votre ordinateur et qui gère les conteneurs et les images Docker. Lorsque vous utilisez l'interface de ligne de commande Docker

(CLI) pour créer, exécuter ou gérer des conteneurs, vous interagissez avec le démon Docker.

Le démon Docker est un composant essentiel de la plateforme qui garantit que les conteneurs peuvent être démarrés et arrêtés automatiquement lorsque le système démarre.

### Client Docker

Le client Docker permet aux utilisateurs d'interagir avec le démon Docker à l'aide de son interface de ligne de commande (CLI). En termes simples, il s'agit de la partie principale de l'architecture Docker pour la création, la gestion et l'exécution d'applications de conteneurs.

Lorsque vous utilisez la CLI de Docker pour transmettre une commande, le client Docker envoie la commande au démon Docker exécuté sur votre ordinateur, qui exécute alors l'opération demandée. Le **client Docker peut être installé** sur n'importe quelle machine devant interagir avec le démon Docker, y compris votre machine locale, un serveur distant ou un serveur virtuel.

Votre serveur d'hébergement  
virtuel, vos règles !

[Obtenir un hébergement VPS](#)



## Conteneur Docker

Un conteneur Docker est un paquet qui contient tous les prérequis nécessaires à l'exécution d'une application.

Les conteneurs sont conçus pour être très portables, ce qui signifie qu'ils peuvent être facilement déplacés d'un environnement à un autre, par exemple de l'ordinateur portable d'un développeur à un environnement de test ou d'un environnement de test à un environnement de production.

## Image Docker

Une image Docker est un modèle préconfiguré qui spécifie ce qui doit être inclus dans un conteneur Docker. En général, les images sont téléchargées à partir de sites web tels que [Docker Hub](#). Cependant, il est également possible de créer une image personnalisée à l'aide de Dockerfile.

## Registre Docker

Le registre Docker est un référentiel central qui stocke et gère les images Docker. Il s'agit d'un système basé sur un serveur qui permet aux utilisateurs de stocker et de partager des images Docker avec d'autres, ce qui facilite la distribution et le déploiement d'applications. Le registre Docker le plus connu est Docker Hub.

# Cheat sheet des commandes Docker

Maintenant que vous savez comment fonctionne Docker, examinons quelques-uns des exemples de commandes Docker les plus populaires.

## Commandes Build

Docker utilise la commande build pour construire des images à partir d'un fichier Docker. Les commandes les plus courantes sont les suivantes :

Commande	Explication
<code>docker build</code>	Construit une image à partir d'un fichier Docker situé dans le répertoire courant.

```
ddocker build
https://github.com/docker/
rootfs.git#container:docker
```

Construit une image à partir d'un dépôt GIT distant

```
docker build -t nomimage/tag
```

Construit et marque une image pour en faciliter le suivi

```
docker build
https://votreserver/file.tar.gz
```

Construit une image à partir d'une archive tar distante

```
docker build -t image:1.0
-<<EOFFFROM busyboxRUN echo
"hello world"EOF
```

Construit une image via un fichier Docker passé par STDIN

## Commandes de nettoyage

Pour garder votre système propre et économiser de l'espace disque, c'est une bonne idée de nettoyer les images, conteneurs et volumes inutilisés. Consultez les commandes ci-dessous pour plus de détails :

### Commande

### Explication

```
docker image prune
```

Efface une image inutilisée

```
docker image prune -a
```

Efface toutes les images qui ne sont pas utilisées par les conteneurs.

```
docker system prune
```

Supprime tous les conteneurs arrêtés, tous les réseaux non utilisés par les conteneurs, toutes les images en suspens et tout le cache de construction.

```
docker image rm image
```

Supprime une image

```
docker rm container
```

Supprime un conteneur en cours d'exécution

<code>docker swarm leave</code>	Laisse un swarm
<code>docker stack rm stackname</code>	Supprime un swarm
<code>docker volume rm \$(docker volume ls -f dangling=true -q)</code>	Supprime tous les volumes en suspens
<code>docker rm \$(docker ps - a -q)</code>	Supprime tous les conteneurs arrêtés
<code>docker kill \$ (docker ps - q)</code>	Arrête tous les conteneurs en cours d'exécution

## Commandes d'interaction avec les conteneurs

Interagissez avec votre conteneur Docker à l'aide des commandes courantes suivantes :

Commande	Explication
<code>docker start container</code>	Démarrage d'un nouveau conteneur
<code>docker stop container</code>	Arrête un conteneur
<code>docker pause container</code>	Met en pause un conteneur
<code>docker unpause container</code>	Désactiver un conteneur
<code>docker restart container</code>	Redémarrage d'un conteneur
<code>docker wait container</code>	Bloque un conteneur

<code>docker export container</code>	Exporte le contenu du conteneur vers une archive tar
<code>docker attach container</code>	S'attache à un conteneur en cours d'exécution
<code>docker wait container</code>	Attend que le conteneur se termine et affiche le code de sortie.
<code>docker commit -m "commit message" -a "author" container username/image_name : tag</code>	Sauvegarde d'un conteneur en cours d'exécution sous forme d'image
<code>docker logs -ft container</code>	Suivi des registres de conteneurs
<code>docker exec -ti container script.sh</code>	Exécute une commande dans un conteneur
<code>docker commit container image</code>	Crée une nouvelle image à partir d'un conteneur
<code>docker create image</code>	Crée un nouveau conteneur à partir d'une image

## Commandes d'inspection des conteneurs

Parfois, vous devez inspecter vos conteneurs à des fins d'assurance qualité ou de dépannage. Ces commandes vous permettent d'avoir une vue d'ensemble de ce que font les différents conteneurs :

Commande	Explication
<code>docker ps</code>	Liste de tous les conteneurs en cours d'exécution

<code>docker -ps -a</code>	Liste de tous les conteneurs
<code>docker diff container</code>	Inspecte les modifications apportées aux répertoires et aux fichiers du système de fichiers du conteneur.
<code>docker top container</code>	Affiche tous les processus en cours dans un conteneur existant
<code>docker inspect container</code>	Affiche des informations de bas niveau sur un conteneur
<code>docker logs container</code>	Rassemble les journaux d'un conteneur
<code>docker stats container</code>	Affiche les statistiques d'utilisation des ressources du conteneur

## Commandes de gestion des images

Parmi les commandes de gestion d'images les plus courantes, on peut citer

Commande	Explication
<code>image docker ls</code>	Listes d'images
<code>docker image rm mysql</code>	Supprime une image
<code>docker tag image tag</code>	Tags d'une image
<code>docker history image</code>	Affiche l'historique de l'image
<code>docker inspect image</code>	Affiche des informations de bas niveau sur une image

## Commandes Run

Docker utilise la commande run pour créer des conteneurs à partir d'images fournies. La syntaxe par défaut de cette commande est la suivante :

```
docker run (options) image (command) (arg...)
```

Après la syntaxe par défaut, utilisez l'un des flags suivants :

Flag	Explication
-detach , -d	Exécute un conteneur en arrière-plan et imprime l'identifiant du conteneur.
-env , -e	Définit les variables d'environnement
-hostname , -h	Attribue un nom d'hôte à un conteneur
-label , -l	Crée une étiquette de métadonnées pour un conteneur
-name	Attribue un nom à un conteneur
-network	Connecte un conteneur à un réseau
-rm	Retire le conteneur lorsqu'il s'arrête
-read-only	Définit le système de fichiers du conteneur en lecture seule
-workdir , -w	Définit un répertoire de travail dans un conteneur



## Commandes du registre

Si vous devez interagir avec Docker Hub, utilisez les commandes suivantes :

Commande	Explication
<code>docker login</code>	Connexion à un registre
<code>docker logout</code>	Déconnexion d'un registre
<code>docker pull mysql</code>	Extrait une image d'un registre
<code>docker push repo/ rhel-httpd:latest</code>	Pousse une image vers un registre
<code>docker search term</code>	Recherche dans Docker Hub les images avec le terme spécifié

## Commandes de service

Gérez tous les services Docker à l'aide de ces commandes de base :

Commande	Explication
<code>docker service ls</code>	Liste tous les services en cours d'exécution dans un swarm
<code>docker stack services stackname</code>	Liste de tous les services en cours d'exécution
<code>docker service ps servicename</code>	Liste des tâches d'un service
<code>docker service update servicename</code>	Mise à jour d'un service

`docker service create image`

Création d'un nouveau service

`docker service scale  
servicename=10`

Mise à l'échelle d'un ou plusieurs  
services répliqués

`docker service logs stackname  
servicename`

Liste de tous les journaux de service

## Commandes de réseau

Si vous devez interagir avec le réseau Docker, utilisez l'une des commandes suivantes :

### Commande

### Explication

`docker network create networkname`

Création d'un nouveau réseau

`docker network rm networkname`

Supprime un réseau spécifié

`docker network ls`

Liste de tous les réseaux

`docker network connect  
networkname container`

Connecte un conteneur à un  
réseau

`docker network disconnect  
networkname container`

Déconnecte un conteneur du  
réseau

`docker network inspect  
networkname`

Affiche des informations détaillées  
sur un réseau

Votre serveur d'hébergement



virtuel, vos règles !

Obtenir un hébergement VPS



## Suggestions de lecture

Consultez nos autres tutoriels Docker :

[Comment installer Docker sur Ubuntu](#)

[Comment installer WordPress sur Docker](#)

[Comment créer un conteneur Docker](#)

## Conclusion

Docker est un outil formidable pour tous ceux qui souhaitent essayer les conteneurs. La courbe d'apprentissage peut être abrupte si vous n'êtes pas familier avec le développement basé sur les conteneurs. Heureusement, le fait d'avoir une antiseiche à portée de main peut accélérer le processus, car toutes les commandes courantes sont facilement accessibles et vous n'avez pas besoin de les chercher sur Internet.

Dans ce tutoriel, nous avons couvert les bases de l'architecture Docker et passé en revue toutes les commandes Docker de base.

Nous espérons que ce tutoriel Docker vous a été utile. Si vous avez des questions, laissez-les dans la section des commentaires ci-dessous.



L'AUTEUR

**Chaimaa Chakir**

Chaimaa est une spécialiste du référencement et du marketing de contenu chez Hostinger. Elle est passionnée par le marketing digital et la technologie. Elle espère aider les gens à résoudre leurs problèmes et à réussir en ligne. Chaimaa est une cinéphile qui adore les chats et l'analyse des films.

[Plus d'informations sur Chaimaa Chakir](#)

Offre Black Friday → 64% DE REMISE

# Hébergement VPS

+

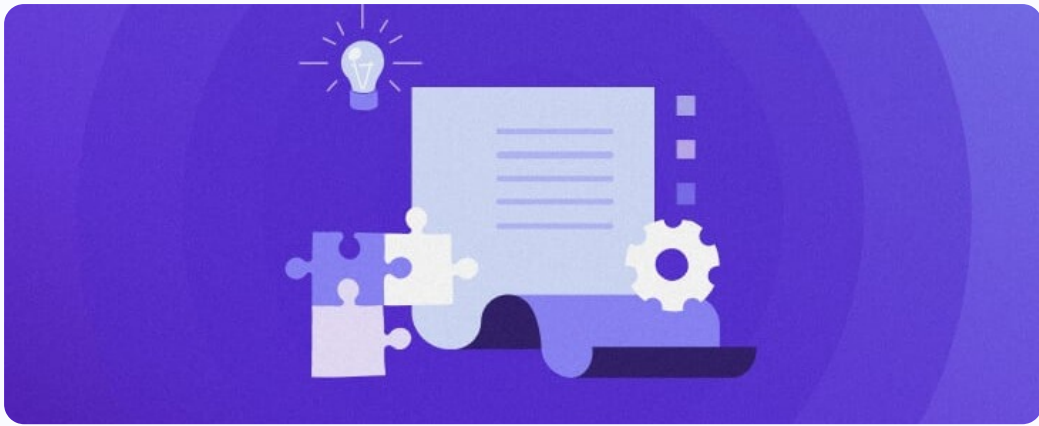
Sauvegardes hebdomadaires  
gratuites

**6.49€**  
/mois

avec un plan de 2 ans

**Obtenir**

**Tutoriels relatifs**

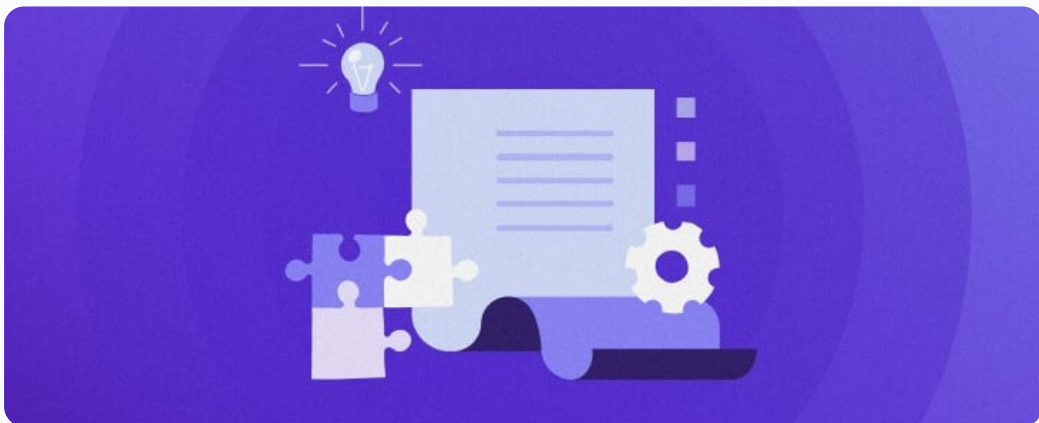


01 Oct • VPS

## Comment créer un serveur Teamspeak : pour Linux, Windows, macOS + pointer le domaine vers le serveur

TeamSpeak 3 est une application de protocole de voix sur IP qui permet aux utilisateurs de communiquer et de collaborer en temps réel à l'aide...

Par Chaimaa Chakir

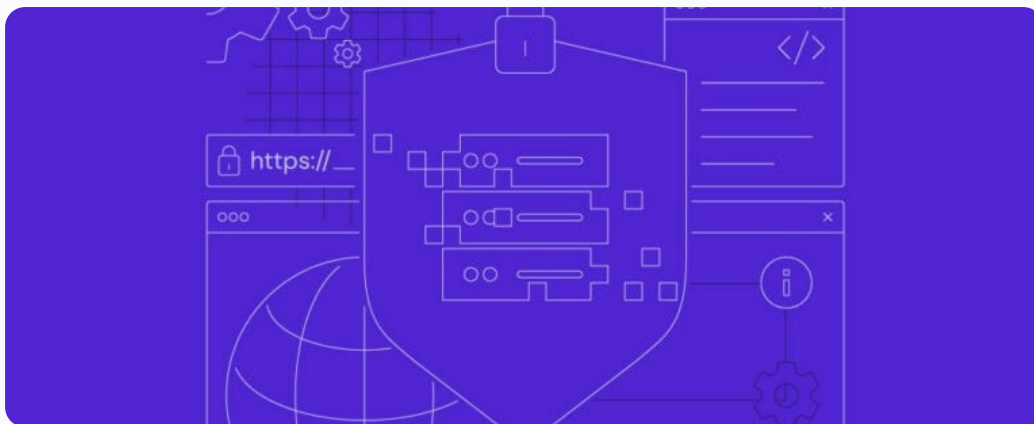


10 Sep • VPS • AUTOMATISATION

## Comment intégrer n8n à Ollama pour les flux de travail LLM locaux ?

L'intégration de n8n à Ollama vous permet d'exploiter divers modèles d'IA dans votre flux de travail automatisé, ce qui lui...

Par Chaimaa Chakir



10 Sep • VPS • AUTOMATISATION

## Comment intégrer n8n à WhatsApp ?

L'intégration de n8n à WhatsApp vous permet de créer un flux de travail automatisé pour diverses tâches, notamment la création d'un...

Par Chaimaa Chakir

## Ce que disent nos clients

Excellent



Basé sur 54 640 avis



# Laissez une réponse

Commentaire\*

Prénom\*

Email\*

- ☐ En utilisant ce formulaire, vous acceptez que vos données personnelles soient traitées conformément à notre **Politique de confidentialité**.

**Soumettre**

## HÉBERGEMENT

Hébergement web

Hébergement pour WordPress

Hébergement VPS

n8n auto-hébergé

Email professionnel

Hébergement Cloud

Hébergement pour WooCommerce

Hébergement pour agences

Hébergement Minecraft

Hébergement de serveurs de jeux

Google Workspace

## NOM DE DOMAINE

[Acheter un nom de domaine](#)

[Noms de domaine pas chers](#)

[Nom de domaine gratuit](#)

[WHOIS](#)

[Certificat SSL Gratuit](#)

[Transfert de nom de domaine](#)

[Extensions de nom de domaine](#)

[Nom de domaine .FR](#)

## OUTILS

[Horizons](#)

[Créateur de sites internet](#)

[Créateur de sites internet IA](#)

[Créateur de boutiques en ligne](#)

[Créateur de noms d'entreprise](#)

[Créateur de logo IA](#)

[Migrer vers Hostinger](#)

[API Hostinger](#)

## INFORMATION

[Tarifs](#)

[Avis](#)

[Programme d'Affiliation](#)

[Programme de parrainage](#)

[Feuille de route](#)

[Mur de la Renommée](#)

[État du système](#)

[Plan du site](#)

## ENTREPRISE

[À propos](#)

[Nos technologies](#)

[Blog \(en anglais\)](#)

[Principes](#)

## SUPPORT



[Tutoriels](#)

[Base de connaissances](#)

[Contact](#)

[Signaler un Abus](#)



[Politique de  
Confidentialité](#)

[Politique de  
remboursement](#)

[Conditions  
d'utilisation](#)



**et autres**

---

© 2004–2025 Hostinger. Donnez vie à vos projets, évoluez et réussissez en ligne grâce à la puissance de l'IA.

Les prix n'incluent pas la TVA