

Présentation Docker

Introduction

Docker est un ensemble d'outils pour conditionner une application en vue de sa publication.

Ce conditionnement est appelé une image de conteneur.

Docker Inc est l'entreprise qui publie cet ensemble d'outils sous license libre ([Apache License 2.0](#)).



Usage

Docker est utilisé pour gérer les environnements

- de développement
- de test
- de production*

(*) l'usage de certains composants de Docker est controversé

Intérêt

- facilite le processus d'on-boarding
- permet la reproductibilité
- permet de simplifier le déploiement

- sécurise les environnement de test
-

Docker Desktop

Pour commencer en douceur, installons [Docker Desktop](#).

Avec cet outil, nous pourrons :

- trouver une image
 - fabriquer une image
 - lancer une image
 - publier une image
-

Notes pour Microsoft Windows

Selon l'installation :

- WSL
 - HyperV
-

Droits d'installation

Il faut avoir accès aux droits administrateur pour pouvoir installer

Droits de gestion

Pour gérer Docker (mises à jour, lancement), il faut que votre utilisateur appartienne au groupe `docker-users`

[Documentation détaillée](#) sur les droits d'accès nécessaires à l'exploitation de Docker en environnement Microsoft Windows

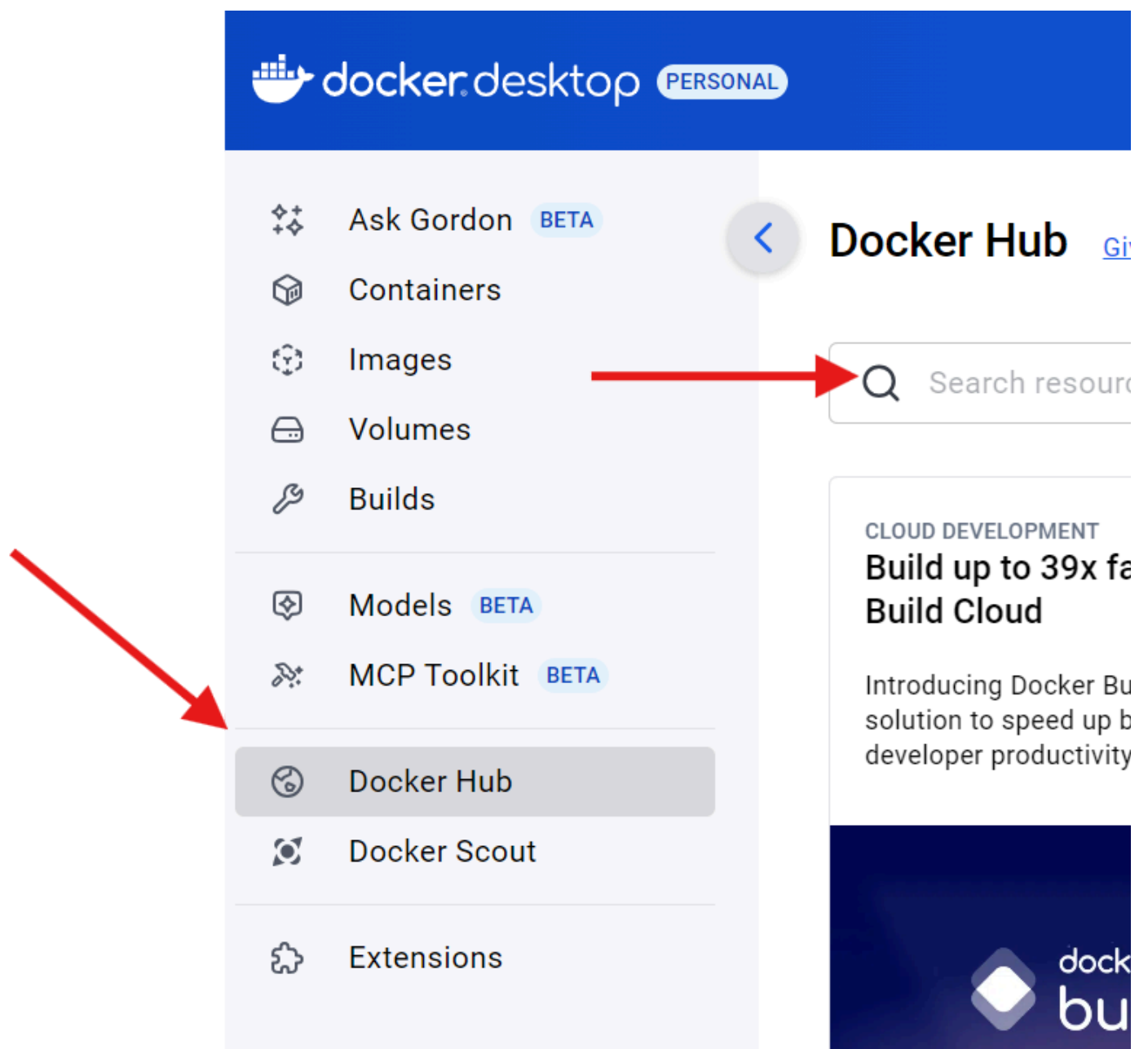
Une image ?

C'est une archive compressée contenant l'ensemble des instructions pour créer un processus dans un environnement isolé.

Cet environnement isolé d'exécution du processus est ce qu'on appelle un conteneur.

Chercher une image

- [Docker Hub sur le web](#)
- ou Docker Desktop



 Ask Gordon **BETA**

 Containers

 Images

 Volumes

 Builds

 Models **BETA**

 MCP Toolkit **BETA**

 **Docker Hub**

 Docker Scout

 Extensions

[Docker Hub](#) / Search

Search results for “alpine

 alpine

1 - 24 of 2500 results



alpine 

A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

 1B+  10K+

 **docker desktop** PERSONAL

Ctrl+K

 Ask Gordon BETA

 Containers

 Images

 Volumes

 Builds

 Models BETA

 MCP Toolkit BETA

 Docker Hub

 Docker Scout

 Extensions

[Docker Hub](#) / alpine



alpine

 Docker Official Image

A minimal Docker image based on Alpine Linux with a complete pac

OPERATING SYSTEMS

☆ 10K+ ↓ 1B+

Overview

Tags

Quick reference

- Maintained by:
[Natanael Copa](#) (an Alpine Linux maintainer)
- Where to get help:
[the Docker Community Slack](#), [Server Fault](#), [Unix & Linux](#), or [Stack Overflow](#)

Ligne de commande

Ouvrez le PowerShell et tapez :

```
docker run alpine
```

Lancer une image locale

Si vous avez fabriqué une image, ou que vous l'avez déjà cherchée, on a pas besoin de passer par le hub :



Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Local

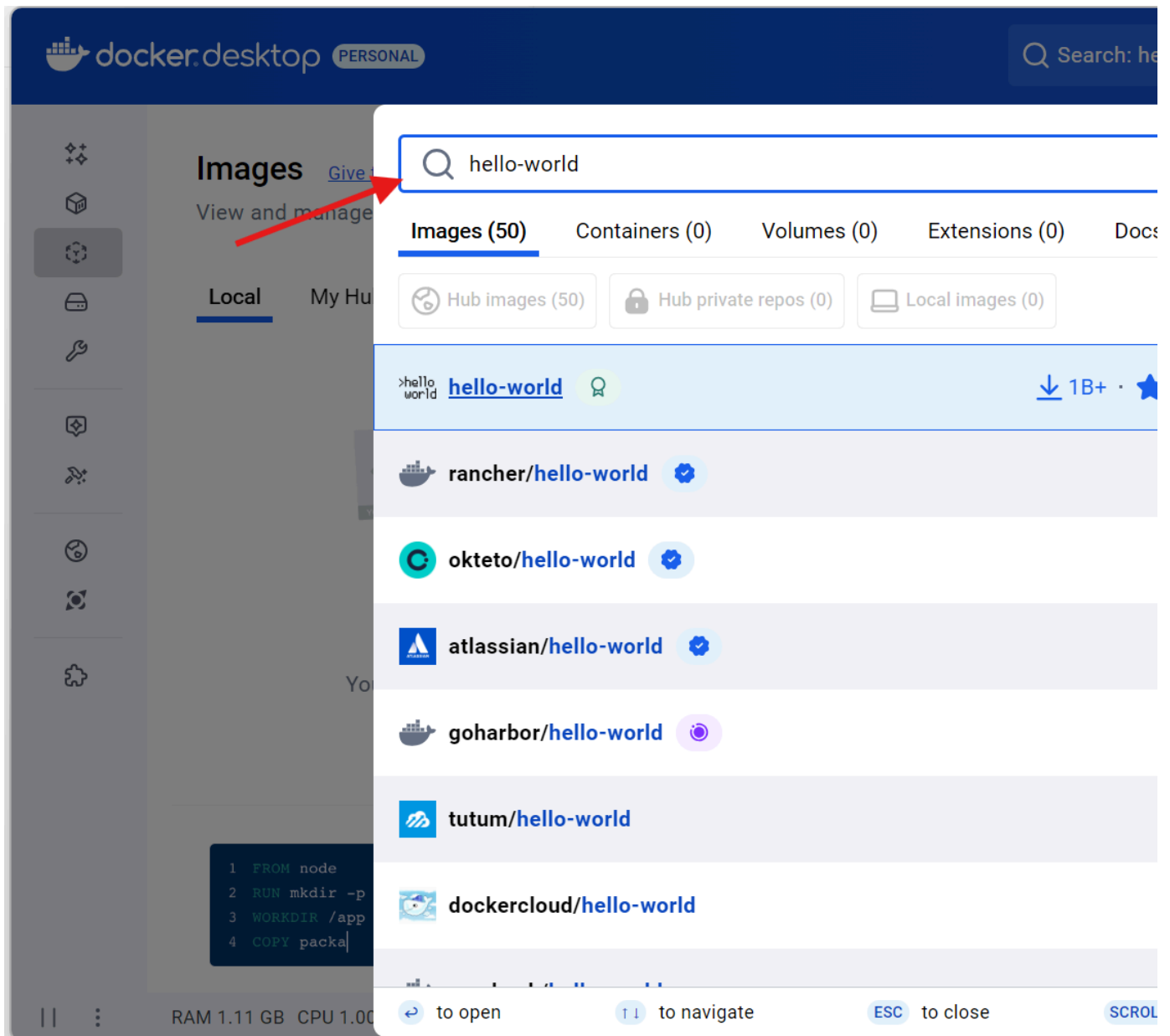
My Hub



Images are used to run containers

You can either build an image from a Dockerfile, or download a pre-built image from Docker Hub.

[Search images to run](#)



Lancer l'image



Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Local My Hub

0 Bytes / 65.51 KB in use 1 images



☐	Name	Tag	Image ID	Created
☐	hello-work	latest	ec153840d1e6	7 months ago

Le nom d'une image

Composé de quatre éléments :

1. l'URL vers le registre
2. le nom du namespace (par défaut `_`)
3. le nom de l'image
4. la version de l'image (par défaut `latest`)

 **Exemple:** `mon-compte/serveur-web:1.1.0`

L'image `serveur-web` dans le namespace `mon-compte` à la version `1.1.0`, fournit par le registre par défaut <https://hub.docker.com>

Docker Hub

Docker Inc met à disposition un hébergement d'images de conteneurs qui s'appelle [Docker Hub](#).

C'est le registre (registry) officiel des images Docker.

Hébergement des images

Le namespace correspond au nom de votre compte.

Le namespace par défaut  correspond aux images produites par Docker Inc.

Si vous avez un compte, un namespace correspondant au nom de votre compte vous sera réservé et vous pourrez y héberger des images.

Warning

Le registre Docker Hub est public. Attention à ne pas y pousser des données confidentielles telles que les mots de passe, le code source de votre client, etc

Il est possible de publier plus qu'une image privée si vous :

- payez
 - déployez votre propre registre : [Docker Registry](#)
 - utilisez un registre tel que celui fournit par [GitLab](#)
-

Lancer un conteneur

Nous allons lancer un serveur MariaDB (une base de données) dans un conteneur et le rendre disponible à notre réseau local.

Depuis Docker Desktop

The screenshot shows the Docker Desktop interface. The top bar is blue with a search icon, a question mark, a bell, a graduation cap, a gear, a grid, and a profile icon 'L'. The sidebar on the left has icons for images, containers, services, volumes, settings, and a gear. The main area is titled 'Images / mariadb:latest'. It shows a green 'IN USE' badge, a 'Recommended fixes' dropdown, and a 'Run' button highlighted by a red arrow. Below the 'Run' button is a 'Vulnerabilities' tab and a 'Packages' tab. The 'Layers (24)' section lists layers 0 through 6 with their sizes. The 'Analyzed by docker scout' section shows a message: 'This image has not been analyzed. You can use Docker Scout to analyze local images and list its vulnerabilities.' with a 'Start analysis' button. The bottom status bar shows 'RAM 1.08 GB CPU 0.13%'.

Images / mariadb:latest

IN USE

Recommended fixes

a5af517d5226

Run

Analyzed by docker scout

Vulnerabilities Packages

Give feedback

Layers (24)

0	A...	0 B
1	A...	0 B
2	L...	0 B
3	L...	0 B
4	A...	87.58 MB
5	C...	0 B
6	R...	53.25 KB

This image has not been analyzed

You can use Docker Scout to analyze local images and list its vulnerabilities.

Start analysis

RAM 1.08 GB CPU 0.13%

Run a new container
mariadb:latest

Optional settings

Container name

A random name is generated if you do not provide one.

Ports
Enter "0" to assign randomly generated host ports.

Host port: 3306 :3306/tcp

Volumes

Host path ... Container path +

Environment variables

Variable: MARIADB_ALLOW_EMPTY_ROOT_PASSWORD Value: Y +

Cancel Run

RAM 1.07 GB CPU 0.25%

En ligne de commande

```
docker run \
  --detach \
  --env MARIADB_ALLOW_EMPTY_ROOT_PASSWORD=Y \
  --publish 3306:3306 \
  mariadb
```

Vous devriez obtenir une sortie similaire à celle-ci :

```
PS C:\Users\mwolff\projets\ma-base> docker run --detach --env MARIADB_ALLOW_EMPTY_ROOT
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
2d8cff3e3cb6: Pull complete
76249c7cd503: Pull complete
420bcadcb59f: Pull complete
ef6def1cca19: Pull complete
5eb5ed4a1ef8: Pull complete
cdac09aaab61: Pull complete
4f6fbbc50826: Pull complete
b9fec364e7a: Pull complete
Digest: sha256:a5af517d52262da1583cebe6b7c970d7bbeaf1f016938c2fc3cf2e95c52d8c08
Status: Downloaded newer image for mariadb:latest
f799a0568cb3e7d37527f83db51390f48c815a5833a6a1dbcff7f80b7908292b
PS C:\Users\mwolff\projets\ma-base>
```

Parce qu'on a demandé à ce que le conteneur soit détaché, `docker run` indique l'identifiant unique du conteneur créé :

```
PS C:\Users\mwolff\projets\ma-base> docker run --detach --env MARIADB_ALLOW_EMPTY_ROOT
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
2d8cff3e3cb6: Pull complete
76249c7cd503: Pull complete
420bcadcb59f: Pull complete
ef6def1cca19: Pull complete
5eb5ed4a1ef8: Pull complete
cdac09aaab61: Pull complete
4f6fbbc50826: Pull complete
b9fec364e7a: Pull complete
Digest: sha256:a5af517d52262da1583cebe6b7c970d7bbeaf1f016938c2fc3cf2e95c52d8c08
Status: Downloaded newer image for mariadb:latest
f799a0568cb3e7d37527f83db51390f48c815a5833a6a1dbcff7f80b7908292b
PS C:\Users\mwolff\projets\ma-base>
```

Nous pouvons retrouver un identifiant court à l'aide de la commande

```
docker ps
```

```
PS C:\Users\mwolff\projets\ma-base> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
f799a0568cb3   mariadb   "docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  0.0.0.0:33
ible_fermat
```

Tip

En général, quand je souhaite manipuler des conteneurs depuis un shell, je conserve leur identifiant dans une variable d'environnement :

```
$Env:cid = "f799a05"
```

Exécuter une commande dans un conteneur

```
docker exec ${Env:cid} whoami
docker exec ${Env:cid} ps ax
docker exec ${Env:cid} hostname -i
docker exec --interactive --tty ${Env:cid} bash
docker exec --interactive --tty ${Env:cid} mariadb mysql
```

Stop! Start!

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

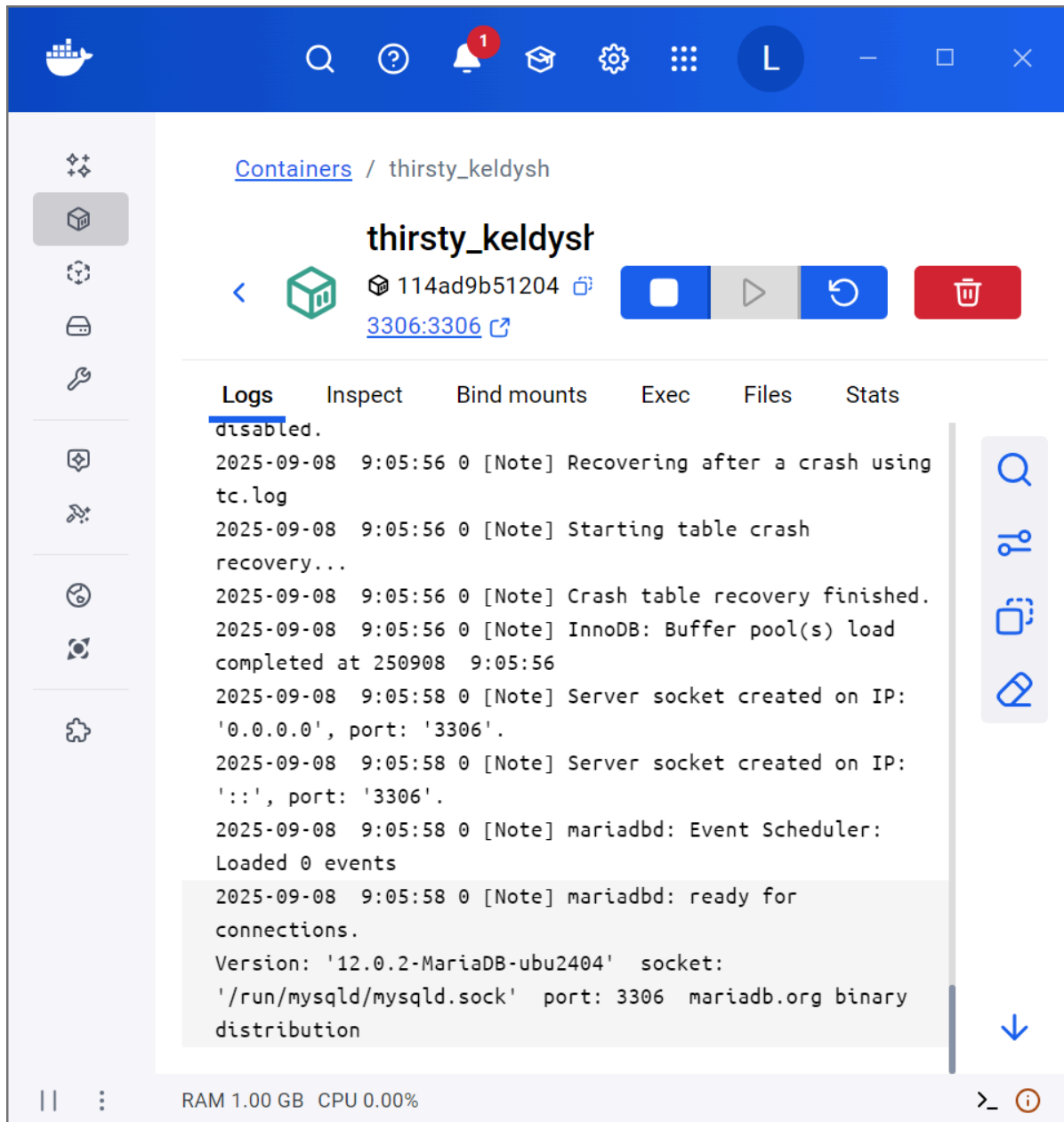
Container CPU usage ⓘ 0.06% / 800% (8 CPUs available) Container memory usage ⓘ 122.2MB / 3.67GB [Show charts](#)

Search Only running

☐	Name	Container ID	Actions
☐	musing_bardeer	d3c0077746	
☐	naughty_swirles	5ca7eebae4	
☐	thirsty_keldysh	114ad9b51	

```
docker stop ${Env:cid}
docker start ${Env:cid}
```

Logs



The screenshot shows the Docker Desktop application window. The top bar is blue with various icons. The left sidebar contains icons for containers, volumes, networks, and settings. The main area displays the 'Containers' tab with a list of containers. The container 'thirsty_keldysh' is selected, showing its ID '114ad9b51204' and port mapping '3306:3306'. Below this, the 'Logs' tab is active, displaying a list of log entries. The log entries show the container's startup sequence, including recovery from a crash, table recovery, and the MariaDB server starting up. The status bar at the bottom indicates 'RAM 1.00 GB' and 'CPU 0.00%'.

Containers / thirsty_keldysh

thirsty_keldysh

<  114ad9b51204     

[3306:3306](#) 

Logs Inspect Bind mounts Exec Files Stats

disabled.

2025-09-08 9:05:56 0 [Note] Recovering after a crash using tc.log

2025-09-08 9:05:56 0 [Note] Starting table crash recovery...

2025-09-08 9:05:56 0 [Note] Crash table recovery finished.

2025-09-08 9:05:56 0 [Note] InnoDB: Buffer pool(s) load completed at 250908 9:05:56

2025-09-08 9:05:58 0 [Note] Server socket created on IP: '0.0.0.0', port: '3306'.

2025-09-08 9:05:58 0 [Note] Server socket created on IP: '::', port: '3306'.

2025-09-08 9:05:58 0 [Note] mariadbd: Event Scheduler: Loaded 0 events

2025-09-08 9:05:58 0 [Note] mariadbd: ready for connections.

Version: '12.0.2-MariaDB-ubu2404' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution

RAM 1.00 GB CPU 0.00%

```
docker logs ${Env:cid}
```

Aller plus loin

- Documentation officielle