

# Docker: Getting started

*How to start using Stackhero for Docker*

## Welcome on Stackhero's documentation!

Stackhero provides Docker instances that are ready for production in just 2 minutes!

Including TLS encryption (aka HTTPS), customizable domains names, backups and updates in just a click.

Try our [managed Docker cloud](#) in just 2 minutes

- [How to use Docker CLI remotely](#)
  - [Install Docker certificates](#)
  - [Run Docker containers remotely](#)
  - [Deploy with Docker-compose](#)

## How to use Docker CLI remotely

Docker CLI should be installed on your computer.

If you haven't installed it yet, you can download it here: <https://www.docker.com/products/docker-desktop>.

With **Stackhero for Docker**, you are able to connect to your Docker server remotely from your computer like you do when you work locally. It will not change your habits and will provide you a very powerful way to handle your containers in production!

To do that, we will use Docker "contexts".

Docker contexts give you the ability to use the Docker CLI on your computer and select where the commands should run: on your local Docker daemon (like usual) or on your **Stackhero for Docker** instance.

### Install Docker certificates

To connect remotely to your **Stackhero for Docker** instance you have to install its certificates first.

These certificates will handle authentication and encryption between your computer and your **Stackhero for Docker** instance.

On your computer, simply copy/paste this command:

```
# STACKHERO_DOCKER_HOST is your Stackhero for Docker instance domain name
# (xxxxxx.stackhero-network.com).
# STACKHERO_DOCKER_SERVICE_ID is your Stackhero service ID.
# STACKHERO_DOCKER_CERTIFICATES_PASSWORD is the password defined in your
Stackhero for Docker configuration.
export STACKHERO_DOCKER_HOST="<STACKHERO_DOCKER_HOST>"
export STACKHERO_DOCKER_SERVICE_ID="<STACKHERO_DOCKER_SERVICE_ID>"
```

```
export STACKHERO_DOCKER_CERTIFICATES_PASSWORD="
<STACKHERO_DOCKER_CERTIFICATES_PASSWORD>"

cd /tmp/ \
  && curl -o certificates.tar
https://docker:${STACKHERO_DOCKER_CERTIFICATES_PASSWORD}@${STACKHERO_DOCKER_HOST}
/stackhero/docker/certificates.tar \
  && tar -xf certificates.tar \
  && (docker context rm -f ${STACKHERO_DOCKER_HOST} 2> /dev/null || true) \
  && docker context create ${STACKHERO_DOCKER_HOST} \
    --description "${STACKHERO_DOCKER_SERVICE_ID} (${STACKHERO_DOCKER_HOST})" \
    --docker
"host=tcp://${STACKHERO_DOCKER_HOST}:2376,ca=ca.pem,cert=cert.pem,key=key.pem"
```

You now have a Docker context that has been created and named with your service domain name. To see all your contexts you can run the command `docker context ls`.

Note that if you update your service domain the certificates will be changed and you'll have to reinstall them.

## Run Docker containers remotely

Right now, if you list your containers running with the command `docker ps`, your Docker CLI will use your local Docker daemon so you will see containers running on your computer.

As an example, you can run this command to get your Docker daemon public IP: `docker run alpine wget -q -O - ifconfig.me`

In that case, as the container runs on your computer, you will see your internet router IP as a reply.

Let's change the Docker context to use your **Stackhero for Docker** instance.

All you have to do is run the command `docker context use <STACKHERO_DOCKER_HOST>`.

Now, Docker commands started on your computer will run on your remote Docker daemon securely!

Run again the same command to retrieve your Docker daemon public IP: `docker run alpine wget -q -O - ifconfig.me`

You should now see your **Stackhero for Docker** instance public IP as a reply, confirming your container is running on your server and no more on your computer!

So now, each Docker command you will start on your computer will be in fact running on your **Stackhero for Docker** instance Docker daemon!

To use your local Docker daemon again, simply change back your Docker context to "default": `docker context use default`

When you mount a volume on a remote container, the data accessible will be the ones on the remote server and not the ones on your computer.

So running a command like `docker run -it -v ${PWD}:/mnt alpine` will not mount the directory from your computer as you can expect.

## Deploy with Docker-compose

As with the Docker CLI, the Docker-compose CLI will use your current context.

So, once you've switched to your remote instance with `docker context use <STACKHERO_DOCKER_HOST>`, every Docker-compose command will run on your remote instance too!

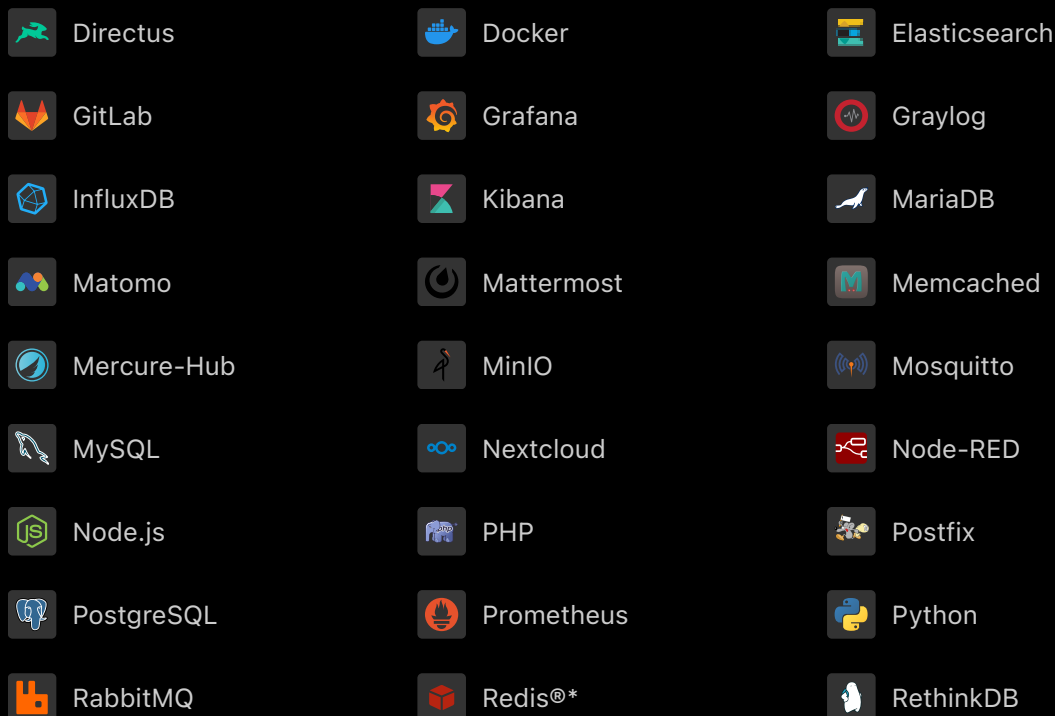
Using `docker context` command is great but is not the good solution when you need to handle contexts in your scripts or Makefiles.

In that case, you should have a look at the `DOCKER_CONTEXT` environment variable that we describe in our advanced documentations pages.

## Other articles about Docker that might interest you

- [Introduction](#)  
Introduction to Stackhero for Docker
- [Advanced context usages](#)  
How to use Docker contexts in your scripts
- [Volumes](#)  
How to persist your containers data with Docker volumes
- [HTTPS routing](#)  
How to handle HTTP routing and TLS encryption (HTTPS) in your Docker containers

### Our Managed Services



[Terms of Service](#)

[Privacy Policy](#)

[Documentations](#)

[Support](#)

[Status](#)

English

Global



Directus, Docker, Elasticsearch, GitLab, Grafana, Graylog, InfluxDB, Kibana, MariaDB, Matomo, Mattermost, Memcached, Mercure-Hub, MinIO, MongoDB, Mosquitto, MySQL, Nextcloud, Node-RED, Node.js, PHP, Postfix, PostgreSQL, Prometheus, Python, RabbitMQ, Redis<sup>®\*</sup>, RethinkDB are trademarks and property of their respective owners. All product and service names used on this website are for identification purposes of their open sourced products only and do not imply endorsement. Stackhero is not affiliated to these trademarks or companies.

\*Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by Stackhero is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and Stackhero

Some icons of this website are made by Dimitry Miroliubov.

© Stackhero. All rights reserved.