Digital Guide

(https://www.ionos.com/digitalguide/)

30.11.22 | Know-how (https://www.ionos.com/digitalguide/server/know-how/)

om/ ners om/ nos

Using Redis in Docker containers

Redis is one of the most popular Docker images. The key-value store makes it possible to save data from Docker containers (https://www.ionos.com/digitalguide/server/know-how/docker-container/). If you want to take advantage of this feature, you can connect Redis and Docker in the **terminal**. Keep reading to find out how.

Contents

Requirements for using Redis

After installing Docker on a Linux server (https://www.ionos.com/digitalguide/server/know-how/installing-and-running-docker-on-a-linux-server/), people often ask themselves where they can save data from Docker containers. Redis (https://www.ionos.com/digitalguide/hosting/technical-matters/what-is-redis/) is a **NoSQL database** that offers an ideal solution, thanks to its lightning-fast performance. Before you connect Redis and Docker, make sure that the following requirements are met:

- Docker is installed (https://www.ionos.com/digitalguide/server/configuration/docker-tutorial-installation-and-first-steps/) and running.
- You're familiar with the basic commands and features of Docker (https://www.ionos.com/digitalguide/server/know-how/what-is-docker/).
- You have a **cloud server** running Linux.

Running Redis in a Docker container

The official Redis image (https://hub.docker.com/_/redis/) is one of the top three most popular Docker images. It contains the command **EXPOSE 6379** (the default Redis port) which makes it automatically available to any linked containers.

To run a Redis instance in a Docker container named **my-redis-container**, use the command:

```
sudo docker run --name my-redis-container -d redis
```

Connecting to Redis running in a Docker container

To connect to a Redis instance from another Docker container, add **--link [Redis container name or ID]:redis** to that container's **docker run** command.

For example, to launch a container named **my-redis-application** from the official CentOS 7 image and link it to the **my-redis-container** container, use the command:

```
sudo docker run --name my-redis-application --link my-redis-container:redis -d centos
```

To connect to a Redis instance from another Docker container with a command-line interface, link the container and specify the host and port with **-h redis -p 6379**.

For example:

```
sudo docker run -it --name my-redis-cli --link my-redis-container:redis --rm redis redis-cli -h re
```

This will connect you to the new container my-redis-cli with a redis-cli connection to the my-redis-container container. Use **[Ctrl] + [P]** and **[Ctrl] + [Q]** to exit this container and return to the command line.

Connect to a Redis container from a remote server

If you wish to connect to a Docker container running Redis from a remote server, you can use Docker's **port forwarding** to access the container with the host server's IP address or domain name.

To use Docker's port forwarding for Redis, add the flag **-p [host port]:6379** to the docker run command.

For example, to set up port forwarding so that you can connect to the container using port 7001 (the standardized TCP (https://www.ionos.com/digitalguide/server/know-how/introduction-to-tcp/)/UDP (https://www.ionos.com/digitalguide/server/know-how/udp-user-datagram-protocol/) port), the docker run command is:

```
sudo docker run --name my-redis-container -p 7001:6379 -d redis
```

You can then switch to another server and access the my-redis-container container with the command:

```
sudo redis-cli -h [host IP or domain name] -p 7001
```

For example, if the host server running the Redis Docker container is IP address 192.168.0.1, you can access the Redis container **from any server** with the command:

```
sudo redis-cli -h 192.168.0.1 -p 7001
```

❗ Note

If you are using a firewall, you will need to allow external access to the relevant port(s); otherwise access will be blocked. To do this, simply go into your firewall settings and **allow access** to the relevant ports.

Loading a Custom redis.conf File

If you have a custom **redis.conf file** where you've already configured Redis, you can load it at container launch by adding the -v flag to the docker run command:

```
-v [path to custom redis.conf file]:/usr/local/etc/redis/redis.conf
```

Simply add the file path of your conf-file to the command. For example, to load the customized file with the path /data/myredis/redis.conf, the command is:

```
sudo docker run --name my-redis-container -v /data/myredis/redis.conf:/usr/local/etc/redis/redis.c
```

30.11.22 | Know-how (https://www.ionos.com/digitalguide/server/know-how/)

Tutorials (https://www.ionos.com/digitalguide/tags/tutorials/)

Related articles

Understanding and managing Docker container volumes

22.03.2022 | Know-how

Software for managing containers makes working with Linux containers significantly easier. You can not only transfer files between containers but also save files from containers. This article shows you how you can effectively manage containers using Docker container volumes, including extensive Docker volumes examples.