How to Install Redis Server Using Docker Container

March 12, 2022 by Hitesh Jethva (798) under Dedicated Server Hosting

0 Comments

Table of Contents

Redis is a free, open-source, and in-memory key-value store that allows data to be stored and accessed at lightning-fast speeds. It can handle large datasets and maintain high availability. Generally, it is used for database, messaging, and caching functions. Running Redis in a Docker container can significantly shorten and simplify the deployment process.

In this post, we will show you how to install Redis using a Docker container.

Step 1 – Install Docker CE

Before starting, Docker CE must be installed on your server. If not installed, you can install it by following the below steps.

First, update your OS, then add the Docker CE repo with the following commands:

```
dnf update -y
```

```
dnf config-manager --add-repo=https://download.docker.com/linux/ce
```

Once the Docker CE repo is created, run the following command to start the installation:

```
dnf install docker-ce -y
```

Once the Docker CE is installed, start the Docker service and enable it to start at system reboot:

```
systemctl start docker
systemctl enable docker
```

Also Read

How to Install and Use Docker on CentOS 8

Step 2 – Download Redis Image

First, you will need to download the Redis image from the Docker Hub registry. You can download it with the following command:

```
docker pull redis
```

You will get the following output:

```
Using default tag: latest
latest: Pulling from library/redis
5eb5b503b376: Pull complete
6530a7ea3479: Pull complete
91f5202c6d9b: Pull complete
9f1ac212e389: Pull complete
82c311187b72: Pull complete
da84aa65ce64: Pull complete
Digest: sha256:0d9c9aed1eb385336db0bc9b976b6b49774aee3d2b9c2788a0d
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
```

Once the Redis image is downloaded, verify the downloaded image using the following command:

```
docker images
```

You will get the following output:

```
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
redis        latest    f1b6973564e9   5 days ago    113MB
```

Step 3 – Create a Redis Container

At this point, the Redis image is downloaded to your local system. You can now start a Redis container using the following command:

```
docker run -it --name redis-container -d redis
```

Once the Redis container is created, you can check it with the following command:

```
docker ps
```

You will get the following output:

```
CONTAINER ID   IMAGE         COMMAND                  CREATED
64163c8ed78d   redis         "docker-entrypoint.s…"   3 seconds ag
```

You can also see the Redis container log using the following command:

```
docker logs redis-container
```

You will get the following output:

```
                _.-
            _.-``__ ''-._
       _.-``    `.  `_.  ''-._           Redis 6.2.6 (00000000/0) 6
   .-`` .-```.  ```\/    _.,_ ''-._
  (    '      ,       .-`  | `,    )     Running in standalone mode
  |`-._`-...-` __...-.``-._|'' _.-'|     Port: 6379
  |    `-._   `._    /     _.-'    |     PID: 1
   `-._    `-._  `-./  _.-'    _.-'
  |`-._`-._    `-.__.-'    _.-'_.-'|
  |    `-._`-._        _.-'_.-'    |           https://redis.io
   `-._    `-._`-.__.-'_.-'    _.-'
  |`-._`-._    `-.__.-'    _.-'_.-'|
  |    `-._`-._        _.-'_.-'    |
   `-._    `-._`-.__.-'_.-'    _.-'
       `-._    `-.__.-'    _.-'
           `-._        _.-'
               `-.__.-'

1:M 01 Feb 2022 07:08:44.564 # WARNING: The TCP backlog setting of
1:M 01 Feb 2022 07:08:44.564 # Server initialized
1:M 01 Feb 2022 07:08:44.564 # WARNING overcommit_memory is set to
1:M 01 Feb 2022 07:08:44.564 * Ready to accept connections
```

Step 4 – Connect to Redis Container

If you want to connect to the Redis container, run the following command:

```
docker exec -it redis-container bash
```

Once you are connected, you will get the following shell:

```
root@64163c8ed78d:/data#
```

Next, run the following command to connect to the Redis command-line interface:

```
root@64163c8ed78d:/data# redis-cli
```

Now, run the following command to check Redis:

```
127.0.0.1:6379> ping
```

If everything is fine, you will get the following output:

```
PONG
```

Now, exit from the Redis container using the following command:

```
127.0.0.1:6379> exit
root@64163c8ed78d:/data# exit
```

Conclusion

In the above guide, we explained how to install Redis on the Docker container. We also explained how to manage and interact with Redis containers. Try Redis on dedicated servers from Atlantic.Net!

Subscribe

< Previous post

Next post >

ALSO ON ATLANTIC.NET

3 years ago • 2 comments

How have cybersecurity threats changed during the pandemic? Early on ...

Installing and ...

7 months ago • 1 comment

Laravel is an open-source PHP programming language framework based on the ...

How to Ins

3 years ago •

In this post, you how to i open-source

Award-Winning Hosting Solutions & Services

Simple, Fast, Reliable Server Hosting

Contact Us To Get Started

Sales: 866-618-3282 Intl: +1-408-335-0825