

Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)

Redis with Raphael De Lio

Let's learn Redis
together![Follow publication](#)

How to run Redis locally in a Docker Container and manage it with Redis Insight and Redis CLI

Raphael De Lio · [Follow](#)

Published in Redis with Raphael De Lio · 8 min read · Aug 8, 2022

114 1

[Twitter](#) | [LinkedIn](#) | [YouTube](#) | [Instagram](#)

This story is also available as an YouTube video. You can watch it [here](#).

Looking to hide highlights?
You can now hide them from
the "≡" menu.
[Okay, got it](#)



My journey with Redis started one story ago. In my previous story, I discussed [10 things you didn't know about Redis](#) and listed many of its powerful capabilities most people are unaware of.

I'm very interested in understanding and experimenting with Redis' capabilities. This is another story of a series where I am documenting my experience with Redis. Join me in this adventure. 😊

[Join Medium with my referral link - Raphael De Lio](#)

Read every story from Raphael De Lio (and thousands of other writers on Medium). Your membership fee directly supports...

[raphaeldelio.medium.com](#)

In this tutorial, you will learn how to run Redis locally alongside Redis Insight to manage its data and how to perform simple CRUD operations.

In addition, we will also see how we can perform the same operations using Redis CLI and how to connect to it from within the container.

Let's put our hands in the fire!

Running Redis Server and Redis Insight

Starting a Redis Server locally is very easy, and running it with Redis Insight is even easier! For running them, we are going to use Docker and although we're running both applications, I will leave the command to run only the server as well:

Only Redis Server

```
docker run -d --name redis-stack-server -p 6379:6379 redis/redis-stack-server:latest
```

Redis Server + Redis Insight

```
docker run -d --name redis-stack -p 6379:6379 -p 8001:8001 redis/redis-stack:latest
```

We will see the image being downloaded, and when it finishes, we can see that the container is running by executing `docker ps` in our terminal and that it's exposing ports 6379 and 8001 as defined by `-p 6739:6379 -p 8001:8001`:

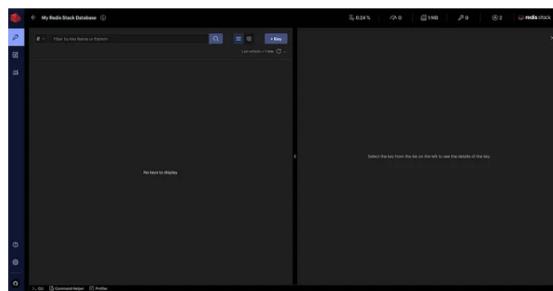
```
Last login: Wed Jul  6 20:36:17 on ttys001
detached from 6739:6379 on ttys001
raphaeldelio@Raphael-MacBook-Pro ~ % docker run -d --name redis-stack-server:latest
Unable to find image "redis/redis-stack-server:latest" locally
latest: Pulling from library/redis-stack-server
d79fe7edba7: Pull complete
187378a1f3c: Pull complete
f91a27092183: Pull complete
4c980e0a2a13: Pull complete
4d774a538d95: Pull complete
c4391bc5b95a: Pull complete
4c27723a0a0e: Pull complete
8ce8d8764f6f: Pull complete
329525a2a2a5: Pull complete
db8ad3732b3: Pull complete
7ec2a2a2a2a5: Pull complete
6f8d8764f653: Pull complete
Digest: sha256:cb099c81c4544353a721f9ea1378a39ff63951a6d783a1374a746b9b729
Status: Downloading [=====>] 100% for redis/redis-stack-server:latest
4c4594a9dc0cf5fd8aaef435024a372c2f2c64fa2d2c5245b22708ae798081
raphaeldelio@Raphael-MacBook-Pro ~ % docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
4c4594a9dc0cf5fd8aaef435024a372c2f2c64fa2d2c5245b22708ae798081

raphaeldelio@Raphael-MacBook-Pro ~ %
```

Redis Insight

6379 is the port of our server and 8001 is the port of Redis Insight. We can connect to it by accessing `localhost:8001`

If we access it, we should see something like:



On the top right, you can see a few stats of the database. On the top left you can check your database details and in the center left you can see your list of key values and on the center-right, you can see the details of these key values.

Remembering that conceptually, Redis is based on the key-value database paradigm. Every piece of data is associated with a key, either directly or indirectly.

Adding a new key-value

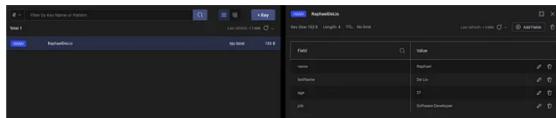
To add a new key value, we need to click on the `+key` button.

Our first key is going to be of type `hash`. The TTL, which is the expiration for the key, is gonna be empty, which means it will never expire.

The key name is going to be `RaphaelDeLio` and then we are going to add some values associated with that key:

Key Type*	Hash	TTL	No limit
Key Name*	RaphaelDeLio		
name	Raphael	<input type="button" value="Delete"/>	<input type="button" value="Add"/>
lastName	De Lio	<input type="button" value="Delete"/>	<input type="button" value="Add"/>
age	27	<input type="button" value="Delete"/>	<input type="button" value="Add"/>
job	Software Developer	<input type="button" value="Delete"/>	<input type="button" value="Add"/>

We can click on `Add Key` and we will see our key in the key list and the fields associated with it:



You can edit or delete any field, or even the key itself, by clicking on the pencils or bins next to them.

Redis CLI

Connecting to Redis CLI

In order to send commands to Redis, we need to connect to Redis CLI. Redis CLI is installed within the container, and we can access it by running the following command in our terminal:

```
docker exec -it redis-stack redis-cli
```

Adding a new key

To add a new key, we are going to use the `HSET` command. This command is specific for setting keys of type `hash`, each type has a different prefix, such as `LSET` for `list`, `SET` for `string`, among others we will see later in this story.

In order to add the same object we added with Redis Insight, we are going to run the following command:

```
HSET RaphaelDeLio name "Raphael" lastName "De Lio" age 27 job  
"Software Developer"
```

As a response, Redis should return the number of fields set in the key:

```
(integer) 4
```

Retrieving a key

There's also a command for getting each type of key and they follow the same pattern we saw before.

We will cover two ways of retrieving hash keys, one is for getting all fields of this hash. We can do it by running:

```
HGETALL RaphaelDeLio
```

And we should see a response like:

```
1) "name"  
2) "Raphael"  
3) "lastName"  
4) "De Lio"  
5) "age"  
6) "27"  
7) "job"  
8) "Software Developer"
```

And the second, is by retrieving an specific field, by running:

```
HGET RaphaelDeLio job
```

Which will return:

```
"Software Developer"
```

Editing a field or adding a new one

Editing a field or adding a new one is done by the same set command `HSET`, we can do it by running:

```
HSET RaphaelDeLio newKey "This is a new Key"  
HSET RaphaelDeLio job "I'm editing an existing key!"
```

And you can test them, by running the GET commands we saw before.

Deleting a field or a key

To delete a hash field, you just need to run the `HDEL` command, as in:

HDEL RaphaelDeLio job

And to delete the key, you run:

DEI_RaphaelDelije

Disconnecting from Redis CLI

To disconnect from Redis CLI just press `ctrl + c`

What about the other types?

Hashes

Redis Hashes are maps between string fields and string values, so they are the perfect data type to represent objects (e.g. A User with a number of fields like name, surname, age, and so forth).

Every hash can store up to $2^{32} - 1$ field-value pairs (more than 4 billion).

String

Strings are the most basic kind of Redis value. Redis Strings are binary safe, this means that a Redis string can contain any kind of data, for instance, a JPEG image or a serialized Ruby object.

A String value can be at a max of 512 Megabytes in length.

Example of a String

Commands for Strings:

- **SET:** To add or edit a key
 - **GET:** To retrieve a key
 - **DEL:** To delete a key

Lists

Redis Lists are simply lists of strings, sorted by insertion order. It is possible to add elements to a Redis List by pushing new elements on the head (on the left) or on the tail (on the right) of the list.

Example of a list

Example of a new element being added to a list

Sets

Redis Sets are an unordered collection of Strings. It is possible to add, remove, and test for the existence of members in O(1).

Redis Sets have the desirable property of not allowing repeated members. Adding the same element multiple times will result in a set having a single copy of this element. Practically speaking this means that adding a member does not require a *check if exists then add* operation.

Creating a set with three elements that are the same

Created set that discarded two of the three elements that were the same

Commands:

- **SADD:** Adds one member to a set
- **SREM:** Removes one member from a set
- **SMEMBERS:** List all members from a set

Sorted Sets

Redis Sorted Sets are, similarly to Redis Sets, non-repeating collections of Strings. The difference is that every member of a Sorted Set is associated with a score, that is used to keep the Sorted Set in order, from the smallest to the greatest score. While members are unique, scores may be repeated.

Example of a sorted set order by the score

Commands:

- **ZADD:** To add new members
- **ZREM:** To remove existing members
- **ZRANGE:** To list members by range

Streams

A Redis stream is a data structure that acts like an append-only log. Streams are useful for recording events in the order they occur.

For Streams, we need to provide an entry key, which can be a timestamp or sequence number. If we leave it as `*`, Redis will assign the Database timestamp to it.

Adding a Stream key

Example of AirQuality being streamed through a period

Commands:

- **XADD**: To append a new entry to a stream
- **XRANGE**: List all entries matching a range of IDs
- **XDEL**: Removes an entry from a stream

JSON

RedisJSON is part of the JSON module and allows you to add JSON objects as the value for your keys.

Example of a JSON Value

Commands:

- **JSON.SET**: To add or replace a new JSON or field in an existing JSON
- **JSON.DEL**: To delete a JSON or field in a JSON
- **JSON.GET**: To retrieve a JSON

Conclusion

In this story, you learned how to start a Redis Server and Redis Insight locally. You also learned about different types of keys supported by Redis and also how to perform simple CRUD operations with Redis Insight and Redis CLI.

I hope you have enjoyed this story. I will be diving more into Redis on the following stories, so don't forget to follow me and subscribe to my next stories!

Contribute

Writing takes time and effort. I love writing and sharing knowledge, but I also have bills to pay. If you like my work, please, consider donating through Buy Me a Coffee: <https://www.buymeacoffee.com/RaphaelDelio>

Or by sending me BitCoin: 1HjG7pmghg3Z8RATH4aiUWr156BGafJ6Zw

Follow Me on Social Media

Stay connected and dive deeper into the world of Redis with me! Follow my journey across all major social platforms for exclusive content, tips, and discussions.

[Twitter](#) | [LinkedIn](#) | [YouTube](#) | [Instagram](#)

 Published in Redis with Raphael De Lio
56 Followers · Last published Feb 24, 2025

Let's learn Redis together!

Follow

 Written by Raphael De Lio
932 Followers · 53 Following

Software Engineer | Developer Advocate | International Conference Speaker | Tech Content Creator | Working @ Redis | <https://linktr.ee/raphaeldelio>

Follow



Responses (1)

 DRucco
What are your thoughts?

 Alirezamosavi
Jul 15, 2023

Hi
Can you please talk about presistence of data in redis stack in docker?

 1 reply 

More from Raphael De Lio and Redis with Raphael De Lio



 In Redis with Raphael De Lio by Raphael De Lio
Can Postgres replace Redis as a cache?

I wanted to understand: Is Postgres as a cache a really good idea? Can it indeed...

Jul 14, 2024  2.2K  31  ...  Jun 12, 2023  192  1  ...



 In Redis with Raphael De Lio by Raphael De Lio
Understanding Redis Streams

Twitter | LinkedIn | YouTube | Instagram



 In Redis with Raphael De Lio by Raphael De Lio
Understanding Persistence in Redis—AOF & RDB + on Docker

Come understand how its mechanisms function and learn how to work with them.

Aug 22, 2022  135  1  ...  Feb 8, 2023  178  1  ...



 In Redis with Raphael De Lio by Raphael De Lio
Understanding Pub/Sub in Redis

Twitter | LinkedIn | YouTube | Instagram This story is also available as a YouTube video....

[See all from Raphael De Lio](#) [See all from Redis with Raphael De Lio](#)

Recommended from Medium





Tihomir Manushev

Replace Redis with PostgreSQL

UNLOGGED Tables + JSONB

Mar 23 310 8

In Coding Beauty by Tari Ibabu

This new IDE from Google is an absolute game changer

This new IDE from Google is seriously revolutionary.

Mar 12 3.3K 194



In Level Up Coding by Renaldi Purwanto

How I Scaled a Go Backend to Handle 1 Million Requests per...

From 100 Requests to 1Million: My Journey in Scaling a Go Backend

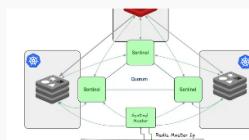
Mar 19 672 14

In Python in Plain English by Dhruv Ahuja

Mastering Background Job Queues with Celery, Redis, and FastAPI

Heads Up! Click here to unlock this article for free if you're not a Medium member!

Oct 12, 2024 227 1



YOUSSEF CHAMRAH

High-Availability Celery on Kubernetes with Redis Sentinel: ...

Introduction When running Celery workers backed by Redis in production, ensuring high...

Dec 7, 2024 97

SaaS Killer



Dipanshu

Paying for software is stupid ..Open-Source tools to...

These 40 Open-Source Tools Will Make Your SaaS Subscriptions Look Obsolete

Mar 27 17K 19

See more recommendations