

Connecting with Python

This guide explains how to connect a Python application to a Redis database using the [redis](#) library. It walks through the required setup, configuration, and execution of a simple Redis command.

Variables

To connect to Redis, the following parameters are needed. You can find these values in the Elestio Redis service overview.

Variable	Description	Purpose
HOST	Redis hostname (from Elestio service overview)	Address of the Redis server.
PORT	Redis port (from Elestio service overview)	Port used to connect to Redis. The default is 6379.
PASSWORD	Redis password (from Elestio service overview)	Authentication credential for the Redis connection.

These values can usually be found in the Elestio service overview details as shown in the image below, make sure to take a copy of these details and add it to the code moving ahead.



redis-aiont

Redis

Cluster

Running

Open terminal

Delete cluster

Add node

Overview

Nodes

Backups

Audit

Termination protection

Disabled. VM can be powered off and terminated.

Protection deactivated



Auto-Failover

Enabled. In case of failure, the cluster will automatically attempt to recover

Auto-Failover activated



Nodes

2 Nodes: 1 Primary, 1 Replica

Add node

Database Admin

Display your database credentials

Hide DB Credentials

Host

redis-aiont-u7774.vm.elestialio.app



Port

26379



User

default



Password

Show password



CLI

redis-cli -h redis-aiont-u7774.vm.elestialio.app -p 26379 --user default --pass '*****'

Show password



Prerequisites

Install Python and pip

- Check if Python is installed by running:

```
python3 --version
```

- If not installed, download and install it from python.org.
- Check pip (Python package installer):

```
pip --version
```

Install the redis Package

Install the official redis library using pip:

```
pip install redis
```

Code

Create a file named `redis.py` and paste the following code:

```
import redis

config = {
    "host": "HOST",
    "port": PORT, # Example: 6379
    "password": "PASSWORD",
    "decode_responses": True
}

try:
    client = redis.Redis(**config)
    client.set("testKey", "Hello Redis")
    value = client.get("testKey")
    print("Connected to Redis")
    print("Retrieved value:", value)

except redis.RedisError as err:
    print("Redis connection or operation failed:", err)
```

To execute the script, open the terminal or command prompt and navigate to the directory where `redis.py` is located. Once in the correct directory, run the script with the command:

```
python3 redis.py
```

If everything is set up correctly, the output will be:

```
Connected to Redis
Retrieved value: Hello Redis
```

Revision #2

Created 19 May 2025 07:00:35 by kaiwalya

Updated 19 May 2025 07:28:41 by kaiwalya