

# How to connect my spring boot app to redis container on docker?

Asked 4 years, 10 months ago   Modified 2 years, 3 months ago   Viewed 6k times



1



I have created a spring app and i want to connect it to redis server which is deployed on docker-compose i put the needed properties as follow : `spring.redis.host=redis` `spring.redis.port=6379`

But i keep getting a `ConnexionException` so how can i Know on which host redis is running and how to connect to it.

Here is my docker-compose file :

```
version: '2'
services:

redis:
  image: 'bitnami/redis:5.0'
  environment:
    # ALLOW_EMPTY_PASSWORD is recommended only for development.
    - ALLOW_EMPTY_PASSWORD=yes
    - REDIS_DISABLE_COMMANDS=FLUSHDB,FLUSHALL
  ports:
    - '6379:6379'
  volumes:
    - 'redis_data:/bitnami/redis/data'
```

volumes: redis\_data: driver: local

```
redis.clients.jedis.exceptions.JedisConnectionException: java.net.UnknownHostException: redis
at redis.clients.jedis.Connection.connect(Connection.java:207) ~[jedis-2.9.0.jar:?]
at redis.clients.jedis.BinaryClient.connect(BinaryClient.java:93) ~[jedis-2.9.0.jar:?]
at redis.clients.jedis.Connection.sendCommand(Connection.java:126) ~[jedis-2.9.0.jar:?]
at redis.clients.jedis.Connection.sendCommand(Connection.java:121) ~[jedis-2.9.0.jar:?]
at redis.clients.jedis.BinaryClient.ping(BinaryClient.java:106) ~[jedis-2.9.0.jar:?]
at redis.clients.jedis.BinaryJedis.ping(BinaryJedis.java:195) ~[jedis-2.9.0.jar:?]
```

spring-boot   docker   redis   docker-compose

Share Edit Follow

edited Feb 13, 2020 at 21:32

asked Feb 13, 2020 at 21:26



Riccardo perreira

47 ● 1 ● 5

Post the error log – [Beez](#) Feb 13, 2020 at 21:28

Check if you can access the Redis instance from where you are running the Spring code. Example - [stackoverflow.com/questions/9866541/...](https://stackoverflow.com/questions/9866541/...) – [Shankar](#) Feb 13, 2020 at 21:35

[Report this ad](#)

## 2 Answers

Sorted by: Highest score (default) 



From docker compose [documentation](#)

3



By default Compose sets up a single network for your app. Each container for a service joins the default network and is both reachable by other containers on that network, and discoverable by them at a hostname identical to the container name



If you want to access redis by container name ('redis' in this case), the Spring boot application has also be deployed as a docker compose service, but it doesn't appear in the docker-compose file that you've provided in the question, so please add it.



Alternatively If you're trying to run the spring boot application on host machine, use 'localhost' instead of 'redis' in order to access the redis container.

Share Edit Follow

answered Feb 13, 2020 at 21:37



[Mark Bramnik](#)

42.3k ● 5 ● 64 ● 107



0

Another approach you can use is **"docker network"** , Below are the steps to follow :

1. Create a docker network for redis  
docker network create redis-docker



## 2. Spin up redis container is redis-docker network.

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

## 3. Inspect the redis-docker container

```
docker network inspect redis-docker
```

```
PS C:\Users\aksha> docker run -d --net redis-docker --name redis-stack -p 6379:6379 -p 8001:8001 redis/redis-stack:latest
f88dd0609126086c32b2784dd895e019529ec1a585af4d5f60d65b8b49f0ec
PS C:\Users\aksha> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
f88dd060912   redis/redis-stack:latest   "/entrypoint.sh"        5 seconds ago   Up 2 seconds   0.0.0.0:6379->6379/tcp, 0.0.0.0:8001->8001/tcp, :::6379->6379/tcp, :::8001->8001/tcp   redis-stack
PS C:\Users\aksha> docker network inspect redis-docker
[
  {
    "Name": "redis-docker",
    "Id": "9f4eed5925d112436d6c3f4a6b6c2d2552d7dda288c8ead23472b33f9af3c87",
    "Created": "2022-09-09T08:00:58.7072869Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.19.0.0/16",
          "Gateway": "172.19.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "f88dd0609126086c32b2784dd895e019529ec1a585af4d5f60d65b8b49f0ec": {
        "Name": "redis-stack",
        "EndpointID": "b3f11e66ef1cd662659fd60c55964a58d162c9d451e73722b178599a622d2408",
        "MacAddress": "02:42:ac:13:00:02",
        "IPv4Address": "172.19.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

## 4. Copy the "IPv4Address" IP and paster in application.yml

```
spring:
  redis: 172.19.0.2
  host:
  password:
  port: 6379
```

## 5. Now **build** , **start** your application.

Share Edit Follow

answered Sep 9, 2022 at 8:10



**Dhruv sahu**

85 ● 1 ● 2