



链滴

redis 镜像使用的正确姿势

作者: [zzm110110](#)

原文链接: <https://ld246.com/article/1567744157333>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.下载redis镜像

```
docker pull redis:latest
```

2.单机模式启动redis

```
docker run --name redis --log-driver local -p 6380:6379 -d -v /usr/local/docker/redis:/data --start=always redis:latest redis-server --appendonly yes --requirepass "123456"
```

3.集群模式启动redis

```
# docker run --name redis7000 -p 7000:6379 -d redis redis-server --requirepass "123456" --appendonly yes --protected-mode no --cluster-enabled yes
```

说明：

redis的密码为123456

Redis持久化功能已启动

保护模式已经关闭

集群模式已开启

我在这里开启6台redis镜像；开启6台redis镜像把宿主机端口和镜像名称改成自己想要的端口和镜像称

4.查看6个redis容器的IP

```
docker inspect redis7000 | grep IPAddress
```

```
[root@liyu123456QWE local]# docker inspect redis7000 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.5",
    "IPAddress": "172.17.0.5",
[root@liyu123456QWE local]# docker inspect redis7001 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.6",
    "IPAddress": "172.17.0.6",
[root@liyu123456QWE local]# docker inspect redis7002 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.7",
    "IPAddress": "172.17.0.7",
[root@liyu123456QWE local]# docker inspect redis7003 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.8",
    "IPAddress": "172.17.0.8",
```

```
[root@liyu123456QWE local]# docker inspect redis7004 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.9",
    "IPAddress": "172.17.0.9",
[root@liyu123456QWE local]# docker inspect redis7005 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.10",
    "IPAddress": "172.17.0.10",
```

5.进入容器查看同一个redis镜像，能否联通其他redis镜像

```
[root@liyu123456QWE local]# docker exec -it redis7000 bash
root@d1430835a5fc:/data# redis-cli
127.0.0.1:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.5 -p 6379
172.17.0.5:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.6 -p 6379
172.17.0.6:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.7 -p 6379
172.17.0.7:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.8 -p 6379
172.17.0.8:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.9 -p 6379
172.17.0.9:6379> exit
root@d1430835a5fc:/data# redis-cli -h 172.17.0.10 -p 6379
172.17.0.10:6379> exit
```

6.最后使用命令，把6台redis镜像集群起来

```
redis-cli -a 123456 --cluster create 172.17.0.5:6379 172.17.0.6:6379 172.17.0.7:6379 172.17.0.8:6379 172.17.0.9:6379 172.17.0.10:6379 --cluster-replicas 1
```

7.集群成功预览

```
root@d1430835a5fc:/data# redis-cli -a 123456 --cluster create 172.17.0.5:6379 172.17.0.6:6379 172.17.0.7:6379 172.17.0.8:6379 172.17.0.9:6379 172.17.0.10:6379 --cluster-replicas 1
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
>>> Performing hash slots allocation on 6 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
Adding replica 172.17.0.9:6379 to 172.17.0.5:6379
Adding replica 172.17.0.10:6379 to 172.17.0.6:6379
Adding replica 172.17.0.8:6379 to 172.17.0.7:6379
M: ee6292668c1c08ddec10a461b0fd098b9ca55c13 172.17.0.5:6379
  slots:[0-5460] (5461 slots) master
M: 33f8ff6b233562141b2ddd038c3e126316fc57fc 172.17.0.6:6379
  slots:[5461-10922] (5462 slots) master
M: a30a6480433c7501f332333b8cbcafd7066d8fde 172.17.0.7:6379
```

```
slots:[10923-16383] (5461 slots) master
S: 02bf188a3e744f8a4efc009d9aee7215bd7d9c39 172.17.0.8:6379
replicates a30a6480433c7501f332333b8cbcafd7066d8fde
S: 1d12e3367d1422eb6fdaeabd82cc4dfde5c5823d 172.17.0.9:6379
replicates ee6292668c1c08ddec10a461b0fd098b9ca55c13
S: 5e919ab4c598f5926cdf480ab69a8e6f3a7e5d3d 172.17.0.10:6379
replicates 33f8ff6b233562141b2ddd038c3e126316fc57fc
Can I set the above configuration? (type 'yes' to accept): yes
>>> Nodes configuration updated
>>> Assign a different config epoch to each node
>>> Sending CLUSTER MEET messages to join the cluster
Waiting for the cluster to join
.....
>>> Performing Cluster Check (using node 172.17.0.5:6379)
M: ee6292668c1c08ddec10a461b0fd098b9ca55c13 172.17.0.5:6379
slots:[0-5460] (5461 slots) master
1 additional replica(s)
M: a30a6480433c7501f332333b8cbcafd7066d8fde 172.17.0.7:6379
slots:[10923-16383] (5461 slots) master
1 additional replica(s)
S: 1d12e3367d1422eb6fdaeabd82cc4dfde5c5823d 172.17.0.9:6379
slots: (0 slots) slave
replicates ee6292668c1c08ddec10a461b0fd098b9ca55c13
S: 5e919ab4c598f5926cdf480ab69a8e6f3a7e5d3d 172.17.0.10:6379
slots: (0 slots) slave
replicates 33f8ff6b233562141b2ddd038c3e126316fc57fc
M: 33f8ff6b233562141b2ddd038c3e126316fc57fc 172.17.0.6:6379
slots:[5461-10922] (5462 slots) master
1 additional replica(s)
S: 02bf188a3e744f8a4efc009d9aee7215bd7d9c39 172.17.0.8:6379
slots: (0 slots) slave
replicates a30a6480433c7501f332333b8cbcafd7066d8fde
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```