Dockerfile 构建 Redis 集群

版权说明

本“ 比特就业课 ”项目（以下简称“本项目”）的所有内容，包括但不限于文字、图片、音频、视频、软件、程序、数据库、设计、布局、界面等，均由本项目的开发者或授权方拥有版权。 我们鼓励个人学习者使用本项目进行学习和研究。在遵守相关法律法规的前提下，个人学习者可以下载、浏览、学习本项目的内容，并为了个人学习、研究或教学目的而使用其中的材料。 但请注意， 未经我们明确授权，个人学习者不得将本项目的内容用于任何商业目的， 包括但不限于销售、转让、许可或以其他方式从中获利。此外，个人学习者也不得擅自修改、复制、传播、展示、表演或制作本项目内容的衍生作品。 任何未经授权的使用均属侵权行为，我们将依法追究法律责任。如果您希望以其他方式使用本项目的内容，包括但不限于引用、转载、摘录、改编等，请事先与我们取得联系，获取书面授权。 感谢您对“比特就业课”项目的关注与支持，我们将持续努力，为您提供更好的学习体验。 特此说明。 比特就业课版权所有方。

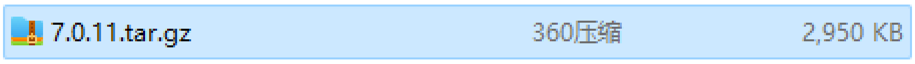
对比特项目感兴趣，可以联系这个微信。

实战目的

通过 dockerfile 和 docker compose 完成 redis 集群的搭建，对 dockerfile 和 compose 有进一步的掌握。

实战步骤

1. 打开 redis 官网 https://redis.io/，找到下载 https://redis.io/download/，下载 7.0.x 对应的版本的源码文件,无法下载的话使用 http://download.redis.io/releases/redis-7.0.11.tar.gz 替换对应版本进行下载



7.0.11.tar.gz 360压缩 2,950 KB

2. 准备目录

```
Shell
mkdir -p /data/maxhou/rediscluster/redis
cd /data/maxhou/rediscluster/redis
wget http://download.redis.io/releases/redis-7.0.11.tar.gz
```

3. 找到里面的配置文件模板，redis.conf,修改以下内容，完成配置文件创建,创建好的文件放到目录/data/maxhou/rediscluster/redis

```
Shell
#表示前台运行
daemonize no
#端口
port 6379
#持久化
dir /data/redis
#启用集群
cluster-enabled yes
#集群参数配置
cluster-config-file nodes.conf
#集群超时时间
cluster-node-timeout 5000
#密码配置
requirepass 123456
#主节点密码配置
masterauth 123456
#表示远端可以连接
bind * -::*
```

4. 在目录/data/maxhou/rediscluster/redis,通过 vi Dockerfile 创建 Dockerfile，用于构建自己的 redis 镜像

```
Shell
FROM centos:7 as buildstage
RUN  sed -e 's|^mirrorlist=|#mirrorlist=|g' \
         -e
's|^#baseurl=http://mirror.centos.org/centos|baseurl=https://mirro
rs.ustc.edu.cn/centos|g' \
         -i.bak \
         /etc/yum.repos.d/CentOS-Base.repo
RUN yum install -y centos-release-scl
RUN yum makecache
RUN yum install -y devtoolset-9-gcc devtoolset-9-gcc-c++
devtoolset-9-binutils make
#wget http://download.redis.io/releases/redis-7.0.11.tar.gz
ADD redis-7.0.11.tar.gz /
ADD redis.conf /redis/
WORKDIR /redis-7.0.11
RUN source /opt/rh/devtoolset-9/enable&& make
RUN mv /redis-7.0.11/src/redis-server /redis/ && mv /redis-
7.0.11/src/redis-cli /redis/
ENTRYPOINT ["/redis/redis-server", "/redis/redis.conf"]


FROM centos:7
RUN mkdir -p /data/redis && mkdir -p /redis
COPY --from=buildstage /redis /redis
EXPOSE 6379
ENTRYPOINT ["/redis/redis-server", "/redis/redis.conf"]
```

5. 通过 docker build -t 测试镜像构建

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster/redis# cd
/data/maxhou/rediscluster/redis
root@139-159-150-152:/data/maxhou/rediscluster/redis#  docker
build -t myredis:v1.0 .
[+] Building 0.1s (17/17) FINISHED
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 905B
0.0s
 => [internal] load .dockerignore
0.1s
 => => transferring context: 2B
```

```
0.0s
 => [internal] load metadata for docker.io/library/centos:7
0.0s
 => [buildstage  1/10] FROM docker.io/library/centos:7
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 74B
0.0s
 => CACHED [stage-1 2/3] RUN mkdir -p /data/redis && mkdir -p
/redis
0.0s
 => CACHED [buildstage  2/10] RUN  sed -e
's|^mirrorlist=|#mirrorlist=|g'          -e
's|^#baseurl=http://mirror.centos.org/centos|baseurl=https://mirro
rs.ustc.edu  0.0s
 => CACHED [buildstage  3/10] RUN yum install -y centos-release-
scl
0.0s
 => CACHED [buildstage  4/10] RUN yum makecache
0.0s
 => CACHED [buildstage  5/10] RUN yum install -y devtoolset-9-gcc
devtoolset-9-gcc-c++ devtoolset-9-binutils make
0.0s
 => CACHED [buildstage  6/10] ADD redis-7.0.11.tar.gz /
0.0s
 => CACHED [buildstage  7/10] ADD redis.conf /redis/
0.0s
 => CACHED [buildstage  8/10] WORKDIR /redis-7.0.11
0.0s
 => CACHED [buildstage  9/10] RUN source /opt/rh/devtoolset-
9/enable&& make
0.0s
 => CACHED [buildstage 10/10] RUN mv /redis-7.0.11/src/redis-
server /redis/ && mv /redis-7.0.11/src/redis-cli /redis/
0.0s
 => CACHED [stage-1 3/3] COPY --from=buildstage /redis /redis
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:694d1bd24530129f45bfb69b372c8944c4df9b14ad9b5116febb5049cbe
```

```
c1307
0.0s
 => => naming to docker.io/library/myredis:v1.0
```

6. 启动一个容器测试是服务能否正常运行,可以看到容器正常运行

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster/redis# docker run -
d --name testmyredis  myredis:v1.0
fd9495f93cec0f5b195df4631e03d22100c8d314634daa6d96dacde6d66a6871
root@139-159-150-152:/data/maxhou/rediscluster/redis# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS             PORTS      NAMES
fd9495f93cec   myredis:v1.0   "/redis/redis-server…"   3 seconds
ago       Up 1 second        6379/tcp   testmyredis
```

7. 清理容器释放资源，因后续要启动的实例较多，要看下服务器资源是否足够

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster/redis# docker rm -f
testmyredis
testmyredis
```

8. 编写 docker-compose.yml

```
Shell
version: "3"
services:
  redis01:
      image: myredis:v1.0
      build: ./redis
      ports:
        - 6379:6379
      container_name: redis01
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis02:
      image: myredis:v1.0
      container_name: redis02
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis03:
      image: myredis:v1.0
      container_name: redis03
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis04:
      image: myredis:v1.0
      container_name: redis04
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis05:
      image: myredis:v1.0
      container_name: redis05
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis06:
      image: myredis:v1.0
      container_name: redis06
      healthcheck:
        test:  /redis/redis-cli ping
        interval: 10s
        timeout: 5s
        retries: 10
  redis07:
      image: myredis:v1.0
      container_name: redis07
      entrypoint: ["/redis/redis-cli","--cluster","create","redis01:6379","redis02:6379","redis03:6379","redis04:6379","redis05:6379","redis06:6379","--cluster-replicas","1","-a","123456","--cluster-yes"]
      depends_on:
        redis01:
         condition: service_healthy
        redis02:
         condition: service_healthy
        redis03:
         condition: service_healthy
        redis04:
         condition: service_healthy
        redis05:
         condition: service_healthy
        redis06:
         condition: service_healthy
```

9. 执行构建可以完成镜像构建，我们之前测试的时候已经构建过了，所以这一步非常快

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster# docker compose
build
[+] Building 0.3s (18/18) FINISHED
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 950B
0.0s
 => [internal] load .dockerignore
0.1s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/centos:7
0.0s
 => [buildstage  1/11] FROM docker.io/library/centos:7
0.0s
 => [internal] load build context
0.2s
 => => transferring context: 3.10MB
0.2s
 => CACHED [stage-1 2/3] RUN mkdir -p /data/redis && mkdir -p
/redis
0.0s
 => CACHED [buildstage  2/11] RUN  sed -e
's|^mirrorlist=|#mirrorlist=|g'          -e
's|^#baseurl=http://mirror.centos.org/centos|baseurl=https://mirro
rs.ustc.edu  0.0s
 => CACHED [buildstage  3/11] RUN yum install -y centos-release-
```

```
scl
0.0s
 => CACHED [buildstage  4/11] RUN yum makecache
0.0s
 => CACHED [buildstage  5/11] RUN yum install -y devtoolset-9-gcc
devtoolset-9-gcc-c++ devtoolset-9-binutils make
0.0s
 => CACHED [buildstage  6/11] ADD redis-7.0.11.tar.gz /
0.0s
 => CACHED [buildstage  7/11] RUN mkdir -p /redis
0.0s
 => CACHED [buildstage  8/11] ADD redis.conf /redis/
0.0s
 => CACHED [buildstage  9/11] WORKDIR /redis-7.0.11
0.0s
 => CACHED [buildstage 10/11] RUN source /opt/rh/devtoolset-
9/enable&& make
0.0s
 => CACHED [buildstage 11/11] RUN mv /redis-7.0.11/src/redis-
server /redis/ && mv /redis-7.0.11/src/redis-cli /redis/
0.0s
 => CACHED [stage-1 3/3] COPY --from=buildstage /redis /redis
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:d5e7cd4aaa50a883f333de2f0ba5add37c060dacd0ba0b0f8ec871c76fb
34d17
0.0s
 => => naming to docker.io/library/myredis:v1.0
0.0s
```

10. 启动服务

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster# docker compose up
-d
[+] Running 8/8
 ✔ Network rediscluster_default  Created
0.1s
 ✔ Container redis06             Healthy
15.7s
```

```
 ✔ Container redis01             Healthy
15.7s
 ✔ Container redis02             Healthy
15.7s
 ✔ Container redis03             Healthy
15.7s
 ✔ Container redis04             Healthy
15.7s
 ✔ Container redis05             Healthy
13.7s
 ✔ Container redis07             Started
16.0s
```

11. 查看容器状态

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster# docker compose ps
-a
NAME                IMAGE               COMMAND
SERVICE             CREATED              STATUS
PORTS
redis01             myredis:v1.0        "/redis/redis-server…"
redis01             About a minute ago   Up About a minute
(healthy)     0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
redis02             myredis:v1.0        "/redis/redis-server…"
redis02             About a minute ago   Up About a minute
(healthy)     6379/tcp
redis03             myredis:v1.0        "/redis/redis-server…"
redis03             About a minute ago   Up About a minute
(healthy)     6379/tcp
redis04             myredis:v1.0        "/redis/redis-server…"
redis04             About a minute ago   Up About a minute
(healthy)     6379/tcp
redis05             myredis:v1.0        "/redis/redis-server…"
redis05             About a minute ago   Up About a minute
(healthy)     6379/tcp
redis06             myredis:v1.0        "/redis/redis-server…"
redis06             About a minute ago   Up About a minute
(healthy)     6379/tcp
redis07             myredis:v1.0        "/redis/redis-cli --…"
redis07             About a minute ago   Exited (0) About a minute
ago
```

12. 查看 07 的日志是否创建成功

```
Shell
root@139-159-150-152:/data/maxhou/rediscluster# docker logs -f
redis07
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 redis05:6379 to redis01:6379
Adding replica redis06:6379 to redis02:6379
Adding replica redis04:6379 to redis03:6379
M: 7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f redis01:6379
   slots:[0-5460] (5461 slots) master
M: 153fd434b56c1f8da6362f12f54f91a45b89152d redis02:6379
   slots:[5461-10922] (5462 slots) master
M: bcc8cc8e524632054616b47c52f1d6524ee557ad redis03:6379
   slots:[10923-16383] (5461 slots) master
S: f68953eedbe97d6c4832c235365121e30d4f746c redis04:6379
   replicates bcc8cc8e524632054616b47c52f1d6524ee557ad
S: 7108278b7fbfe83bdd31d5759c7cad3451797bf0 redis05:6379
   replicates 7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f
S: cb3066944b937456f2e0bc8e898a6c5a7c939ab9 redis06:6379
   replicates 153fd434b56c1f8da6362f12f54f91a45b89152d
>>> 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 redis01:6379)
M: 7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f redis01:6379
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
M: 153fd434b56c1f8da6362f12f54f91a45b89152d 192.168.192.4:6379
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
S: cb3066944b937456f2e0bc8e898a6c5a7c939ab9 192.168.192.3:6379
   slots: (0 slots) slave
   replicates 153fd434b56c1f8da6362f12f54f91a45b89152d
S: 7108278b7fbfe83bdd31d5759c7cad3451797bf0 192.168.192.2:6379
   slots: (0 slots) slave
   replicates 7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f
```

```
S: f68953eedbe97d6c4832c235365121e30d4f746c 192.168.192.7:6379
   slots: (0 slots) slave
   replicates bcc8cc8e524632054616b47c52f1d6524ee557ad
M: bcc8cc8e524632054616b47c52f1d6524ee557ad 192.168.192.6:6379
   slots:[10923-16383] (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

13. 进入任意一个 01-06 的容器检查功能是否正常

```Shell
[root@37807e9bd5aa /]# /redis/redis-cli -c -a 123456
Warning: Using a password with '-a' or '-u' option on the command
line interface may not be safe.
127.0.0.1:6379> cluster info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:6
cluster_size:3
cluster_current_epoch:6
cluster_my_epoch:2
cluster_stats_messages_ping_sent:531
cluster_stats_messages_pong_sent:559
cluster_stats_messages_meet_sent:1
cluster_stats_messages_sent:1091
cluster_stats_messages_ping_received:559
cluster_stats_messages_pong_received:532
cluster_stats_messages_received:1091
total_cluster_links_buffer_limit_exceeded:0
127.0.0.1:6379> cluster nodes
7108278b7fbfe83bdd31d5759c7cad3451797bf0 192.168.192.2:6379@16379
slave 7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f 0 1684151959000 1
connected
7a085f2bba9d2b7b49eae6f4fcc8bf19b9b3609f 192.168.192.5:6379@16379
master - 0 1684151960922 1 connected 0-5460
f68953eedbe97d6c4832c235365121e30d4f746c 192.168.192.7:6379@16379
slave bcc8cc8e524632054616b47c52f1d6524ee557ad 0 1684151960421 3
```

```
connected
bcc8cc8e524632054616b47c52f1d6524ee557ad 192.168.192.6:6379@16379
master - 0 1684151959920 3 connected 10923-16383
cb3066944b937456f2e0bc8e898a6c5a7c939ab9 192.168.192.3:6379@16379
myself,slave 153fd434b56c1f8da6362f12f54f91a45b89152d 0
1684151960000 2 connected
153fd434b56c1f8da6362f12f54f91a45b89152d 192.168.192.4:6379@16379
master - 0 1684151959000 2 connected 5461-10922
127.0.0.1:6379> set a 123
-> Redirected to slot [15495] located at 192.168.192.6:6379
OK
192.168.192.6:6379> get a
"123"
192.168.192.6:6379> exit
[root@37807e9bd5aa /]# exit
exit
```

14. 释放资源

```
Shell
docker compose down
```