

## 1. 安装gcc和ruby

```
[root@localhost ~]# yum install gcc ruby
```

## 2. 到官网下载最新版本

## 3. 将文件解压并复制到创建的目录中

```
[root@localhost ~]# tar -zxvf redis-5.0.5.tar.gz
[root@localhost ~]# mv redis-5.0.5 /usr/local/redis
```

## 4. 编译安装

```
[root@localhost ~]# cd /usr/local/redis/
[root@localhost redis]# make && make install
```

## 5. 创建目录

```
[root@localhost redis]# mkdir -p logs
[root@localhost redis]# mkdir -p redis-cluster/{7000,7001,7002}
```

## 6. 配置

### ◦ 7000端口的配置

```
[root@localhost redis]# cd /usr/local/redis/redis-cluster/7000 && vi redis.conf
```

```
# 端口7000,7001,7002, 与目录对应
port 7000
# 默认ip为127.0.0.1, 需要改为其他节点机器可访问的ip, 否则创建集群时无法访问对应的端口, 无法创建集群
bind 192.168.0.122
# redis后台运行
daemonize yes
# 开启集群
cluster-enabled yes
# 集群的配置, 配置文件首次启动自动生成 7000, 7001, 7002
cluster-config-file nodes.7000.conf
# 请求超时, 默认15秒, 可自行设置
cluster-node-timeout 8000
# 开启aof持久化模式, 每次写操作请求都追加到appendonly.aof文件中
appendonly yes
# 每次有写操作的时候都同步
appendfsync always
# redis服务日志
logfile "/usr/local/redis/logs/redis.log"
# pidfile文件对应7000, 7001, 7002
pidfile /var/run/redis_7000.pid
```

### ◦ 7001端口的配置

```
[root@localhost 7000]# cd /usr/local/redis/redis-cluster/7001 && vi redis.conf
```

```
# 端口7000,7001,7002, 与目录对应
port 7001
# 默认ip为127.0.0.1, 需要改为其他节点机器可访问的ip, 否则创建集群时无法访问对应的端口, 无法创建集群
bind 192.168.0.122
# redis后台运行
daemonize yes
# 开启集群
cluster-enabled yes
# 集群的配置, 配置文件首次启动自动生成 7000, 7001, 7002
cluster-config-file nodes.7001.conf
# 请求超时, 默认15秒, 可自行设置
cluster-node-timeout 8000
# 开启aof持久化模式, 每次写操作请求都追加到appendonly.aof文件中
appendonly yes
# 每次有写操作的时候都同步
appendfsync always
# redis服务日志
logfile "/usr/local/redis/logs/redis.log"
# pidfile文件对应7000, 7001, 7002
pidfile /var/run/redis_7001.pid
```

### ◦ 7002端口的配置

```
[root@localhost 7001]# cd /usr/local/redis/redis-cluster/7002 && vi redis.conf
```

```
# 端口7000,7001,7002, 与目录对应
port 7002
# 默认ip为127.0.0.1, 需要改为其他节点机器可访问的ip, 否则创建集群时无法访问对应的端口, 无法创建集群
bind 192.168.0.122
# redis后台运行
daemonize yes
```

```
# 开启集群
cluster-enabled yes
# 集群的配置，配置文件首次启动自动生成 7000, 7001, 7002
cluster-config-file nodes_7002.conf
# 请求超时，默认15秒，可自行设置
cluster-node-timeout 8000
# 开启aof持久化模式，每次写操作请求都追加到appendonly.aof文件中
appendonly yes
# 每次有写操作的时候都同步
appendfsync always
# redis服务日志
logfile "/usr/local/redis/logs/redis.log"
# pidfile文件对应7000, 7001, 7002
pidfile /var/run/redis_7002.pid
```

## 7. 创建启动和关闭脚本

### ◦ 启动脚本

```
[root@localhost 7002]# cd ~ && vi start-redis.sh

for((i=0;i<3;i++));
do /usr/local/bin/redis-server /usr/local/redis/redis-cluster/700$i/redis.conf;
done
```

### ◦ 关闭脚本

```
[root@localhost ~]# vi stop-redis.sh

for((i=0;i<=2;i++));
do /usr/local/bin/redis-cli -c -h 192.168.0.122 -p 700$i shutdown;
done
```

### ◦ 赋予执行权限

```
[root@localhost ~]# chmod +x start-redis.sh stop-redis.sh
```

## 8. 启动

```
[root@localhost ~]# ./start-redis.sh
```

## 9. 测试

```
[root@localhost ~]# redis-cli -h 192.168.0.122 -p 7000
[root@localhost ~]# redis-cli -h 192.168.0.122 -p 7001
[root@localhost ~]# redis-cli -h 192.168.0.122 -p 7002
```

## 10. 在三台服务器上都启动服务

## 11. 在任意一台机器上执行集群操作

```
[root@localhost ~]# cd /usr/local/redis/src/
[root@localhost src]# [root@localhost src]# ./redis-cli --cluster create 192.168.0.120:7000 192.168.0.120:7001 192.168.0.120:7002 192.168.0.121:30:7000
```

---