# Redis集群的搭建(linux)

Redis集群中至少应该有三个节点。要保证集群的高可用，需要每个节点有一个备份机。

Redis集群至少需要6台服务器。

搭建伪分布式。可以使用一台虚拟机运行6个redis实例。需要修改redis的端口号7001-7006

## 集群搭建环境

1、使用ruby脚本搭建集群。需要ruby的运行环境。

安装ruby

yum install ruby

yum install rubygems

1. 安装ruby脚本运行使用的包。

[root@localhost ~]# gem install redis-3.0.0.gem

Successfully installed redis-3.0.0

1 gem installed

Installing ri documentation for redis-3.0.0...

Installing RDoc documentation for redis-3.0.0...

[root@localhost ~]#

[root@localhost ~]# cd redis-3.0.0/src

[root@localhost src]# ll \*.rb

-rwxrwxr-x. 1 root root 48141 Apr 1 2015 redis-trib.rb

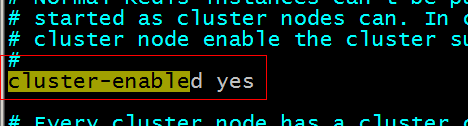
## 搭建步骤

需要6台redis服务器。搭建伪分布式。

需要6个redis实例。

需要运行在不同的端口7001-7006

第一步：创建6个redis实例，每个实例运行在不同的端口。需要修改redis.conf配置文件。配置文件中还需要把cluster-enabled yes前的注释去掉。



第二步：启动每个redis实例。

第三步：使用ruby脚本搭建集群。

|  |
| --- |
| ./redis-trib.rb create --replicas 1 192.168.25.153:7001 192.168.25.153:7002 192.168.25.153:7003 192.168.25.153:7004 192.168.25.153:7005 192.168.25.153:7006 |

创建关闭集群的脚本：

[root@localhost redis-cluster]# vim shutdow-all.sh

redis01/redis-cli -p 7001 shutdown

redis01/redis-cli -p 7002 shutdown

redis01/redis-cli -p 7003 shutdown

redis01/redis-cli -p 7004 shutdown

redis01/redis-cli -p 7005 shutdown

redis01/redis-cli -p 7006 shutdown

[root@localhost redis-cluster]# chmod u+x shutdow-all.sh

|  |
| --- |
| [root@localhost redis-cluster]# ./redis-trib.rb create --replicas 1 192.168.25.153:7001 192.168.25.153:7002 192.168.25.153:7003 192.168.25.153:7004 192.168.25.153:7005 192.168.25.153:7006  >>> Creating cluster  Connecting to node 192.168.25.153:7001: OK  Connecting to node 192.168.25.153:7002: OK  Connecting to node 192.168.25.153:7003: OK  Connecting to node 192.168.25.153:7004: OK  Connecting to node 192.168.25.153:7005: OK  Connecting to node 192.168.25.153:7006: OK  >>> Performing hash slots allocation on 6 nodes...  Using 3 masters:  192.168.25.153:7001  192.168.25.153:7002  192.168.25.153:7003  Adding replica 192.168.25.153:7004 to 192.168.25.153:7001  Adding replica 192.168.25.153:7005 to 192.168.25.153:7002  Adding replica 192.168.25.153:7006 to 192.168.25.153:7003  M: 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3 192.168.25.153:7001  slots:0-5460 (5461 slots) master  M: 8cd93a9a943b4ef851af6a03edd699a6061ace01 192.168.25.153:7002  slots:5461-10922 (5462 slots) master  M: 2935007902d83f20b1253d7f43dae32aab9744e6 192.168.25.153:7003  slots:10923-16383 (5461 slots) master  S: 74f9d9706f848471583929fc8bbde3c8e99e211b 192.168.25.153:7004  replicates 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3  S: 42cc9e25ebb19dda92591364c1df4b3a518b795b 192.168.25.153:7005  replicates 8cd93a9a943b4ef851af6a03edd699a6061ace01  S: 8b1b11d509d29659c2831e7a9f6469c060dfcd39 192.168.25.153:7006  replicates 2935007902d83f20b1253d7f43dae32aab9744e6  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 192.168.25.153:7001)  M: 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3 192.168.25.153:7001  slots:0-5460 (5461 slots) master  M: 8cd93a9a943b4ef851af6a03edd699a6061ace01 192.168.25.153:7002  slots:5461-10922 (5462 slots) master  M: 2935007902d83f20b1253d7f43dae32aab9744e6 192.168.25.153:7003  slots:10923-16383 (5461 slots) master  M: 74f9d9706f848471583929fc8bbde3c8e99e211b 192.168.25.153:7004  slots: (0 slots) master  replicates 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3  M: 42cc9e25ebb19dda92591364c1df4b3a518b795b 192.168.25.153:7005  slots: (0 slots) master  replicates 8cd93a9a943b4ef851af6a03edd699a6061ace01  M: 8b1b11d509d29659c2831e7a9f6469c060dfcd39 192.168.25.153:7006  slots: (0 slots) master  replicates 2935007902d83f20b1253d7f43dae32aab9744e6  [OK] All nodes agree about slots configuration.  >>> Check for open slots...  >>> Check slots coverage...  [OK] All 16384 slots covered.  [root@localhost redis-cluster]# |

# 集群的使用方法(linux)

Redis-cli连接集群。

[root@localhost redis-cluster]# redis01/redis-cli -p 7002 -c

-c：代表连接的是redis集群