目录

[1. 环境准备](#_Toc31757_WPSOffice_Level1) [2](#_Toc31757_WPSOffice_Level1)

[(1) 关闭防火墙](#_Toc10617_WPSOffice_Level2) [2](#_Toc10617_WPSOffice_Level2)

[(2) redis服务系统参数配置](#_Toc8557_WPSOffice_Level2) [2](#_Toc8557_WPSOffice_Level2)

[(3) 关闭selinux](#_Toc30002_WPSOffice_Level2) [2](#_Toc30002_WPSOffice_Level2)

[(4) 安装编译软件gcc，请自行配置yum源](#_Toc24881_WPSOffice_Level2) [2](#_Toc24881_WPSOffice_Level2)

[(5) 下载安装包](#_Toc22715_WPSOffice_Level2) [2](#_Toc22715_WPSOffice_Level2)

[(6) 解压并进去目录](#_Toc11063_WPSOffice_Level2) [2](#_Toc11063_WPSOffice_Level2)

[(7) 编译，安装](#_Toc1380_WPSOffice_Level2) [2](#_Toc1380_WPSOffice_Level2)

[2. 安装方式](#_Toc10617_WPSOffice_Level1) [3](#_Toc10617_WPSOffice_Level1)

[2.1. redis单机](#_Toc19244_WPSOffice_Level2) [3](#_Toc19244_WPSOffice_Level2)

[(1) 配置path路径](#_Toc10617_WPSOffice_Level3) [3](#_Toc10617_WPSOffice_Level3)

[(2) 修改配置文件](#_Toc8557_WPSOffice_Level3) [3](#_Toc8557_WPSOffice_Level3)

[(3) 启动服务](#_Toc30002_WPSOffice_Level3) [3](#_Toc30002_WPSOffice_Level3)

[(4) 验证](#_Toc24881_WPSOffice_Level3) [3](#_Toc24881_WPSOffice_Level3)

[2.2. redis集群](#_Toc24920_WPSOffice_Level2) [3](#_Toc24920_WPSOffice_Level2)

[(1) 下载ruby(在集群一个几点执行即可)](#_Toc22715_WPSOffice_Level3) [3](#_Toc22715_WPSOffice_Level3)

[(2) 解压并进去目录(在集群一个几点执行即可)](#_Toc11063_WPSOffice_Level3) [3](#_Toc11063_WPSOffice_Level3)

[(3) 编译安装(在集群一个几点执行即可)](#_Toc1380_WPSOffice_Level3) [3](#_Toc1380_WPSOffice_Level3)

[(4) 修改path路径](#_Toc19244_WPSOffice_Level3) [3](#_Toc19244_WPSOffice_Level3)

[(5) 创建集群目录和相应的配置文件](#_Toc24920_WPSOffice_Level3) [3](#_Toc24920_WPSOffice_Level3)

[(6) 集群目录结构](#_Toc31478_WPSOffice_Level3) [4](#_Toc31478_WPSOffice_Level3)

[(7) 启动服务](#_Toc8616_WPSOffice_Level3) [5](#_Toc8616_WPSOffice_Level3)

[(8) 创建集群，期间需要输入yes确认集群配置](#_Toc31517_WPSOffice_Level3) [5](#_Toc31517_WPSOffice_Level3)

[(9) 验证集群](#_Toc19855_WPSOffice_Level3) [6](#_Toc19855_WPSOffice_Level3)

[(10) redis使用帮助](#_Toc9084_WPSOffice_Level3) [6](#_Toc9084_WPSOffice_Level3)

# 环境准备

1. 关闭防火墙

systemctl stop firewalld ; systemtl disable firewalld

1. redis服务系统参数配置

echo "vm.overcommit\_memory = 1" >> /etc/sysctl.conf && sysctl -p

echo "echo never > /sys/kernel/mm/transparent\_hugepage/enabled" >> /etc/rc.local

1. 关闭selinux

sed -i '/SELINUX/s/enforcing/disabled/g' /etc/selinux/config && reboot

1. 安装编译软件gcc，请自行配置yum源

yum -y install gcc

1. 下载安装包

cd && wget <http://download.redis.io/releases/redis-4.0.11.tar.gz>

1. 解压并进去目录

tar xf redis-4.0.11.tar.gz && cd redis-4.0.11

1. 编译，安装

make PREFIX=/redis install

# 安装方式

## redis单机

1. 配置path路径

sed -i '$aexport PATH=${PATH}:/redis/bin' /etc/profile && source /etc/profile

1. 修改配置文件

cd /root/redis-4.0.11 && vim redis.conf

* bind 0.0.0.0
* requirepass Hy123456

创建redis服务

cat >> /usr/lib/systemd/system/redis.service <<EOF

[Unit]

Description=Redis

After=network.target

[Service]

ExecStart=/usr/local/redis/src/redis-server /usr/local/redis/redis.conf --daemonize no

ExecStop=/usr/local/redis/src/redis-cli -h 127.0.0.1 -p 6379 shutdown

[Install]

WantedBy=multi-user.target

EOF

1. 启动服务

systemctl start redis

1. 验证

redis-cli -h 127.0.0.1 -a Hy123456

## redis集群

除非特殊说明，集群所有节点都要做[环境准备](#_环境准备)和后续操作

1. 下载ruby(在集群一个几点执行即可)

cd && wget <https://cache.ruby-lang.org/pub/ruby/2.5/ruby-2.5.1.tar.gz>

1. 解压并进去目录(在集群一个几点执行即可)

tar xf ruby-2.5.1.tar.gz && cd ruby-2.5.1

1. 编译安装(在集群一个几点执行即可)

./configure --prefix=/usr/local/ruby && make && make install

1. 修改path路径

sed -i '$aexport PATH=${PATH}:/redis/bin:/usr/local/ruby/bin' /etc/profile && source /etc/profile

1. 创建集群目录和相应的配置文件

cd /redis && mkdir -p conf data/{6379,6380} log

touch conf/{redis.conf,redis-6379.conf,redis-6380.conf}

1. 集群目录结构

[root@node1 /]# tree /redis

/redis

├── bin

│   ├── redis-benchmark

│   ├── redis-check-aof

│   ├── redis-check-rdb

│   ├── redis-cli

│   ├── redis-sentinel -> redis-server

│   └── redis-server

├── conf

│   ├── redis-6379.conf

│   ├── redis-6380.conf

│   └── redis.conf

├── data

│   ├── 6379

│   └── 6380

└── log

6 directories, 9 files

[root@node1 conf]# cat redis.conf

cluster-enabled yes

cluster-node-timeout 5000

appendonly yes

daemonize yes

protected-mode no

[root@node1 conf]# cat redis-6379.conf

include /redis/conf/redis.conf

port 6379

cluster-config-file nodes-6379.conf

pidfile /var/run/redis-6379.pid

dir /redis/data/6379

dbfilename dump-6379.rdb

appendfilename "appendonly-6379.aof"

logfile /redis/log/redis-6379.log

[root@node1 conf]# cat redis-6380.conf

include /redis/conf/redis.conf

port 6380

cluster-config-file nodes-6380.conf

pidfile /var/run/redis-6380.pid

dir /redis/data/6380

dbfilename dump-6380.rdb

appendfilename "appendonly-6380.aof"

logfile /redis/log/redis-6380.log

1. 启动服务

[root@node1 conf]# redis-server redis-6379.conf

[root@node1 conf]# redis-server redis-6380.conf

1. 创建集群，期间需要输入yes确认集群配置

[root@node1 conf]# redis-trib.rb create --replicas 1 192.168.100.11:6379 192.168.100.12:6379 192.168.100.13:6379 192.168.100.11:6380 192.168.100.12:6380 192.168.100.13:6380

>>> Creating cluster

>>> Performing hash slots allocation on 6 nodes...

Using 3 masters:

192.168.100.11:6379

192.168.100.12:6379

192.168.100.13:6379

Adding replica 192.168.100.12:6380 to 192.168.100.11:6379

Adding replica 192.168.100.13:6380 to 192.168.100.12:6379

Adding replica 192.168.100.11:6380 to 192.168.100.13:6379

M: 796ec99aad94e190cd3a6508df2c73c201e05c0b 192.168.100.11:6379

slots:0-5460 (5461 slots) master

M: 2f81f6bfcfb20314061398a9b1ab1784499931fc 192.168.100.12:6379

slots:5461-10922 (5462 slots) master

M: f83c528cdf7252422ad0ba0677badb4262bfc6ee 192.168.100.13:6379

slots:10923-16383 (5461 slots) master

S: ef2020ca53f77f4b5ed834423820601ba8d18367 192.168.100.11:6380

replicates f83c528cdf7252422ad0ba0677badb4262bfc6ee

S: c32bc81c2ce7ef90e086a6b9c0e82f5ae73c9d1c 192.168.100.12:6380

replicates 796ec99aad94e190cd3a6508df2c73c201e05c0b

S: 396d9fca8479b548ebc2960e0d5878e677f59bdb 192.168.100.13:6380

replicates 2f81f6bfcfb20314061398a9b1ab1784499931fc

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 192.168.100.11:6379)

M: 796ec99aad94e190cd3a6508df2c73c201e05c0b 192.168.100.11:6379

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: 396d9fca8479b548ebc2960e0d5878e677f59bdb 192.168.100.13:6380

slots: (0 slots) slave

replicates 2f81f6bfcfb20314061398a9b1ab1784499931fc

S: ef2020ca53f77f4b5ed834423820601ba8d18367 192.168.100.11:6380

slots: (0 slots) slave

replicates f83c528cdf7252422ad0ba0677badb4262bfc6ee

M: f83c528cdf7252422ad0ba0677badb4262bfc6ee 192.168.100.13:6379

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: c32bc81c2ce7ef90e086a6b9c0e82f5ae73c9d1c 192.168.100.12:6380

slots: (0 slots) slave

replicates 796ec99aad94e190cd3a6508df2c73c201e05c0b

M: 2f81f6bfcfb20314061398a9b1ab1784499931fc 192.168.100.12:6379

slots:5461-10922 (5462 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.

1. 验证集群

[root@node1 /]# redis-cli -h 192.168.100.12 -p 6379 cluster nodes | egrep '(master|slave)'

396d9fca8479b548ebc2960e0d5878e677f59bdb 192.168.100.13:6380@16380 slave 2f81f6bfcfb20314061398a9b1ab1784499931fc 0 1538066637421 6 connected

2f81f6bfcfb20314061398a9b1ab1784499931fc 192.168.100.12:6379@16379 myself,master - 0 1538066638000 2 connected 5461-10922

ef2020ca53f77f4b5ed834423820601ba8d18367 192.168.100.11:6380@16380 slave f83c528cdf7252422ad0ba0677badb4262bfc6ee 0 1538066639000 4 connected

796ec99aad94e190cd3a6508df2c73c201e05c0b 192.168.100.11:6379@16379 master - 0 1538066637522 1 connected 0-5460

f83c528cdf7252422ad0ba0677badb4262bfc6ee 192.168.100.13:6379@16379 master - 0 1538066638427 3 connected 10923-16383

c32bc81c2ce7ef90e086a6b9c0e82f5ae73c9d1c 192.168.100.12:6380@16380 slave 796ec99aad94e190cd3a6508df2c73c201e05c0b 0 1538066639437 5 connected

[root@node1 ~]# redis-cli -c -p 6379

127.0.0.1:6379> set k v

-> Redirected to slot [7629] located at 192.168.100.11:6379

OK

192.168.100.11:6379> get k

"v"

192.168.100.11:6379> exit

1. redis使用帮助

https://redis.io/topics/cluster-tutorial