Guide for Setting Up Redis Cluster

12/2/2017

Contents

[1. Hardware specifications 1](#_Toc435806710)

[2. Gather Information 1](#_Toc435806711)

[3. Before you start 1](#_Toc435806712)

[3.1. Files you’ll need 1](#_Toc435806713)

[3.2. Tools you’ll need 1](#_Toc435806714)

[4. Install Prerequisites 1](#_Toc435806715)

[4.1. Update host name 1](#_Toc435806716)

[4.2. Mount CentOS 7.1 Installation DVD 1](#_Toc435806717)

[4.3. Install prerequisites 1](#_Toc435806718)

[5. Install Redis 1](#_Toc435806719)

[5.1. Get Tarball ready 1](#_Toc435806720)

[5.2. Copy Sentinel startup script and config file 1](#_Toc435806721)

[5.3. Extract from Tarball 1](#_Toc435806722)

[5.4. Make Redis Executables 1](#_Toc435806723)

[5.5. Build up Redis folder structure 1](#_Toc435806724)

[6. Setup firewall 1](#_Toc435806725)

[6.1. Update firewall configuration 1](#_Toc435806726)

[6.2. Check firewall status 1](#_Toc435806727)

[7. Configure Redis 1](#_Toc435806728)

[7.1. Configure Redis boot script 1](#_Toc435806729)

[7.2. Configure sentinel config file 1](#_Toc435806730)

[7.3. Configure Redis master node 1](#_Toc435806731)

[7.4. Configure Redis slave nodes 1](#_Toc435806732)

[7.5. Turn Redis into a system service and start after boot 1](#_Toc435806733)

[8. Check Redis installation 1](#_Toc435806734)

[9. Check Sentinel status 1](#_Toc435806735)

# 硬件参数

|  |  |  |
| --- | --- | --- |
| **Item** | **Specification** | **Comments** |
| RAM | 64GB |  |
| CPU | 8 Cores |  |
| OS Disk | 20 GB |  |
| Log Disk | 127 GB | /var/log/redis |
| Data Disk | 200 GB | /var/lib/redis |
| Swap Disk | 128 GB |  |

# 信息收集

Fill the form below

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Key** | **Value** | **Comments** |
| Redis Node 1 | IP Address |  | Master node |
| Redis Node 2 | IP Address |  | Slave node |
| Redis Node 3 | IP Address |  | Slave node |
| App Server 1 | IP Address |  |  |
| App Server 2 | IP Address |  |  |

# 开始之前

## 所需文件

请确保你获得以下文件:

* ubuntu-16.04.1-server-amd64.iso
* redis-3.2.6.tar.gz
* sentinel
* sentinel.conf

## 所需工具

请准备好以下工具:

* WinSCP (<http://winscp.net/> )
* PuTTY (<http://www.putty.org/>)
* Notepad ++ (可选, <https://notepad-plus-plus.org/>) (WinSCP和Notepad++ 的集成可参见 <https://winscp.net/eng/docs/integration_editor>)

# 安装前置

本说明以以下假设为前提: 服务器满足硬件参数条件; Linux HyperV虚拟机集成服务或者VmWare虚拟机的VmWare Tools已安装.

关于Linux集成服务的更多信息请参阅:

<http://blogs.technet.com/b/virtualization/archive/2015/05/01/linux-integration-services-4-0-announcement.aspx>

## 更新Hostname

hostnamectl set-hostname **RedisNodeName**

例如:

hostnamectl set-hostname Redis\_Node\_1

## 装载Ubuntu16.04安装DVD

将DVD插入光驱, 或者使用VmWare或者Hyper-V管理工具加载虚拟ISO文件(ubuntu-16.04.1-server-amd64.iso)

mkdir /media/cdrom

mount /dev/cdrom /media/cdrom

## 安装前置

安装 open ssh：sudo apt-get install openssh-server

辑配置文件，允许以 root 用户通过 ssh 登录：

sudo vi /etc/ssh/sshd\_config

找到：PermitRootLogin prohibit-password禁用

添加：PermitRootLogin yes

sudo service ssh restart

安装后弹出DVD

# 安装Redis

## 准备 Tarball

使用WinSCP上传“redis-3.2.6.tar.gz” 到路径**~**

## 拷贝Sentinel startup文件和配置文件

cp sentinel /etc/init.d/sentinel

cp sentinel.conf /etc/redis/sentinel.conf

## 解压Tarball

cd

tar zxvf redis-3.2.6.tar.gz

## 编译Redis前准备

如果系统没安装build-essential：

sudo apt-get install build-essential

安装**tcl**

wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz

sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/

cd /usr/local/tcl8.6.1/unix/

sudo ./configure

sudo make

sudo make install

## 编译Redis可执行文件

cd

tar zxvf redis-3.2.6.tar.gz

cd redis-3.2.6

cd deps

make hiredis lua jemalloc linenoise

cd ..

make

make test

test运行过程中不应该出现任何错误.

## 建立Redis文件夹结构

cp src/redis-server /usr/local/bin

cp src/redis-cli /usr/local/bin

cp src/redis-sentinel /usr/local/bin

mkdir /etc/redis

mkdir /var/log/redis/

mkdir /var/lib/redis

cp utils/redis\_init\_script /etc/init.d/redis

cp redis.conf /etc/redis/6379.conf

cd ..

rm -v -f redis-3.0.5.tar.gz

rm -v -r -f redis-3.0.5

# 建立防火墙

## 更新防火墙配置

apt install firewalld

firewall-cmd --permanent --zone=internal --add-port=6379/tcp

firewall-cmd --permanent --zone=internal --add-port=16379/tcp

firewall-cmd --permanent --zone=internal --add-port=26379/tcp

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_1\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_2\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_3\_ip\_addrk**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip1**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip2**

firewall-cmd --reload

## 检查防火墙状态

firewall-cmd --zone=internal --list-all

输出应该如下:

internal

interfaces:

sources: ***ip\_addresses***

services: dhcpv6-client ipp-client mdns samba-client ssh

ports: **16379/tcp 6379/tcp 26379/tcp**

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

# 配置Redis

## 配置Redis启动脚本

编辑 “/etc/init.d/redis”, 替换如下:

#!/bin/sh

# chkconfig: 2345 10 90

# description: Start and Stop **redis**

PATH=/usr/local/bin

REDISPORT=6379

EXEC=/usr/local/bin/**redis**-server

**REDIS**\_CLI=/usr/local/bin/**redis**-cli

PIDFILE=/var/run/**redis**.pid

CONF="/etc/**redis**/**6379**.conf"

case "$1" in

start)

if [ -f $PIDFILE ]

then

echo "$PIDFILE exists, process is already running or crashed."

else

echo "Starting **Redis** server..."

$EXEC $CONF

fi

if [ "$?"="0" ]

then

echo "**Redis** is running..."

fi

;;

stop)

if [ ! -f $PIDFILE ]

then

echo "$PIDFILE exists, process is not running."

else

PID=$(cat $PIDFILE)

echo "Stopping..."

$**REDIS**\_CLI -p $REDISPORT SHUTDOWN

while [ -x $PIDFILE ]

do

echo "Waiting for **Redis** to shutdown..."

sleep 1

done

echo "**Redis** stopped"

fi

;;

restart|force-reload)

${0} stop

${0} start

;;

\*)

echo "Usage: /etc/init.d/**redis** {start|stop|restart|fore-reload}"

exit 1

esac

## 配置 Redis master节点

在master节点上,编辑 “/etc/redis/6379.conf”

daemonize yes

pidfile /var/run/redis\_6379.pid

tcp-keepalive 60

logfile "/var/log/redis/redis.log"

dir /var/lib/redis

maxmemory-policy noeviction

appendonly yes

## 配置Redis slave节点

在slave节点上, 编辑“/etc/redis/6379.conf”

daemonize yes

pidfile /var/run/redis\_6379.pid

tcp-keepalive 60

logfile "/var/log/redis/redis.log"

dir /var/lib/redis

slaveof master\_node\_ip\_addr 6379

slave-serve-stale-data no

maxmemory-policy noeviction

appendonly yes

## 将Redis添加到开机启动服务

cd /etc/init.d/

添加脚本的执行权限  
sudo chmod a+x /etc/init.d/redis  
设置开机自动启动  
sudo update-rc.d redis defaults

# 验证Redis安装

reboot

系统重启后, 检查Redis log

cat /var/log/redis/redis.log

输出应该没有错误和警告.

1153:M 20 Nov 13:59:05.733 \* Increased maximum number of open files to 10032 (it was originally set to 1024).

\_.\_

\_.-``\_\_ ''-.\_

\_.-`` `. `\_. ''-.\_ Redis 3.0.5 (00000000/0) 64 bit

.-`` .-```. ```\/ \_.,\_ ''-.\_

( ' , .-` | `, ) Running in standalone mode

|`-.\_`-...-` \_\_...-.``-.\_|'` \_.-'| Port: 6379

| `-.\_ `.\_ / \_.-' | PID: 1153

`-.\_ `-.\_ `-./ \_.-' \_.-'

|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|

| `-.\_`-.\_ \_.-'\_.-' | http://redis.io

`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'

|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|

| `-.\_`-.\_ \_.-'\_.-' |

`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'

`-.\_ `-.\_\_.-' \_.-'

`-.\_ \_.-'

`-.\_\_.-'

1153:M 20 Nov 13:59:05.745 # Server started, Redis version 3.0.5

1153:M 20 Nov 13:59:05.745 \* The server is now ready to accept connections on port 6379

Slave节点上输出应为

1189:S 20 Nov 14:09:28.872 \* Connecting to MASTER 192.168.100.211:6379

1189:S 20 Nov 14:09:28.873 \* MASTER <-> SLAVE sync started

# 配置 Sentinel

使用WinSCP上传“sentinel”到路径“/etc/init.d/”

使用WinSCP上传“sentinel.conf”到路径“/etc/redis/”

## 配置sentinel配置文件

在所有的redis服务器上, 按需要编辑“/etc/redis/sentinel.conf”

**sentinel monitor mymaster redis\_node\_1\_ip\_addr 6379 2**

**sentinel down-after-milliseconds mymaster 5000**

**sentinel config-epoch mymaster 0**

**sentinel leader-epoch mymaster 0**

**min-slaves-to-write 1**

**min-slaves-max-lag 10**

**daemonize yes**

**pidfile "/var/run/redis-sentinel.pid"**

**loglevel verbose**

**logfile "/var/log/redis/sentinel.log"**

**sentinel parallel-syncs mymaster 1**

## 将 Sentinel添加到开机启动服务

cd /etc/init.d

添加脚本的执行权限  
chmod a+x sentinel  
设置开机自动启动  
sudo update-rc.d sentinel defaults

## 启动 Sentinel服务

service sentinel start

## 检查Sentinel状态

redis-cli –p 26379

在命令行中输入:

sentinel master mymaster

输入应该如下, 并且没有错误和警告.

1) "name"

2) "mymaster"

3) "ip"

4) "127.0.0.1"

5) "port"

6) "6379"

7) "runid"

8) "953ae6a589449c13ddefaee3538d356d287f509b"

9) "flags"

10) "master"

11) "link-pending-commands"

12) "0"

13) "link-refcount"

14) "1"

15) "last-ping-sent"

16) "0"

17) "last-ok-ping-reply"

18) "735"

19) "last-ping-reply"

20) "735"

21) "down-after-milliseconds"

22) "5000"

23) "info-refresh"

24) "126"

25) "role-reported"

26) "master"

27) "role-reported-time"

28) "532439"

29) "config-epoch"

30) "1"

31) "num-slaves"

32) "1"

33) "num-other-sentinels"

34) "2"

35) "quorum"

36) "2"

37) "failover-timeout"

38) "60000"

39) "parallel-syncs"

40) "1"

## Redis Security

Redis provides a tiny layer of authentication that is optionally turned on editing the **redis.conf** file.

When the authorization layer is enabled, Redis will refuse any query by unauthenticated clients. A client can authenticate itself by sending the **AUTH** command followed by the password.

The password is set by the system administrator in clear text inside the redis.conf file. It should be long enough to preent brute force attacks.

To set the Redis password, edit your redis.conf file, find this line:

# requirepass foobared

Then uncomment it and change foobared to your password. Mkae sure you choose something pretty long, 32 characters or so would probably be good:

requirepass 10d9a99851a411cdae8c3fa09d7290df192441a9

After setting the password, save the file, and restart Redis:

$ sudo service redis-server restart

To check that Redis is working, use the Redis command line. The **redis-cli** command is used to access the Redis command line:

redis-cli

If you already set a password for Redis, you have to auth after connecting:

127.0.0.1:6379> auth your\_redis\_password