Redis cluster 部署

作者：aliezHub
原文链接：https://ld246.com/article/1589782832402
来源网站：链滴

环境准备：

| 类别 | IP| 规格|主机名|

| 1| 192.168.43.120| 4C8G |redis-cluster-01|

| 2| 192.168.43.121| 4C8G |redis-cluster-02|

| 3| 192.168.43.122| 4C8G |redis-cluster-03|

注意：系统均为centos7.6，均已关闭firewalld和selinux。

集群架构是三台机器，每台机器部署两个节点，端口分别是6379和6380.

第一步：下载Redis离线二进制包

```
# wget http://download.redis.io/releases/redis-4.0.10.tar.gz
```

第二步：解压

```
# tar xf redis-4.0.10.tar.gz
# mv redis-4.0.10 /usr/local/redis
```

第三步：编译&&编译安装

```
# cd /usr/local/redis
# make
# make install
```

第四步：将二进制可执行文件拷贝至/usr/local/bin或者/usr/bin目录下

```
# cd src/
# cp redis-trib.rb /usr/local/bin/
```

第五步：检查

```
# cd ../utils/
# ls -l
总用量 76
-rw-rw-r-- 1 root root 593 6月 13 2018 build-static-symbols.tcl
-rw-rw-r-- 1 root root 1303 6月 13 2018 cluster_fail_time.tcl
-rw-rw-r-- 1 root root 1070 6月 13 2018 corrupt_rdb.c
drwxrwxr-x 2 root root 4096 6月 13 2018 create-cluster
-rwxrwxr-x 1 root root 2137 6月 13 2018 generate-command-help.rb
drwxrwxr-x 3 root root 4096 6月 13 2018 graphs
drwxrwxr-x 2 root root 4096 6月 13 2018 hashtable
drwxrwxr-x 2 root root 4096 6月 13 2018 hyperloglog
-rwxrwxr-x 1 root root 9567 6月 13 2018 install_server.sh
drwxrwxr-x 2 root root 4096 6月 13 2018 lru
-rw-rw-r-- 1 root root 1277 6月 13 2018 redis-copy.rb
-rwxrwxr-x 1 root root 1352 6月 13 2018 redis_init_script
-rwxrwxr-x 1 root root 1047 6月 13 2018 redis_init_script.tpl
-rw-rw-r-- 1 root root 1762 6月 13 2018 redis-sha1.rb
drwxrwxr-x 2 root root 4096 6月 13 2018 releasetools
-rwxrwxr-x 1 root root 3787 6月 13 2018 speed-regression.tcl
-rwxrwxr-x 1 root root 693 6月 13 2018 whatisdoing.sh
```

第六步：部署节点1

```
# ./install_server.sh
Welcome to the redis service installer This script will help you easily set up a running redis ser
er
Please select the redis port for this instance: [6379] Selecting default: 6379
Please select the redis config file name [/etc/redis/6379.conf] /data/redis/conf/6379.conf
Please select the redis log file name [/var/log/redis_6379.log] /data/redis/logs/6379.log
Please select the data directory for this instance [/var/lib/redis/6379] /data/redis/data/6379 P
ease select the redis executable path [/usr/local/bin/redis-server]
Selected config: Port : 6379
Config file : /data/redis/6379.conf
Log file : /data/redis/logs/6379.log
Data dir : /data/redis/6379
Executable : /usr/local/bin/redis-server
Cli Executable : /usr/local/bin/redis-cli
Is this ok?
Then press ENTER to go on or Ctrl-C to abort.
Copied /tmp/6379.conf => /etc/init.d/redis_6379
Installing service... Successfully added to chkconfig!
Successfully added to runlevels 345!
Starting Redis server... Installation successful!
```

注意：

①.如果是部署单机Redis，到这里就算结束了。

②.由于部署的集群是单机双实例（如果有条件的话，可以一台机器只部署一个节点），所以第六步和七步都要执行。

第七步：部署节点2

```
# ./install_server.sh
Welcome to the redis service installer This script will help you easily set up a running redis ser
er
```

```
Please select the redis port for this instance: [6379] 6380
Please select the redis config file name [/etc/redis/6380.conf] /data/redis/conf/6380.conf
Please select the redis log file name [/var/log/redis_6380.log] /data/redis/logs/6380.log
Please select the data directory for this instance [/var/lib/redis/6380] /data/redis/data/6380 P
ease select the redis executable path [/usr/local/bin/redis-server]
Selected config: Port : 6380
Config file : /data/redis/6380.conf
Log file : /data/redis/logs/6380.log
Data dir : /data/redis/6380
Executable : /usr/local/bin/redis-server
Cli Executable : /usr/local/bin/redis-cli
Is this ok?
Then press ENTER to go on or Ctrl-C to abort.
Copied /tmp/6380.conf => /etc/init.d/redis_6380
Installing service... Successfully added to chkconfig!
Successfully added to runlevels 345!
Starting Redis server... Installation successful!
```

第八步：检查

```
# tree -L 3
.
└── redis
    ├── 6379
    ├── 6379.conf
    ├── 6380
    ├── 6380.conf
    └── logs
        ├── 6379.log
        └── 6380.log
4 directories, 4 files
```

另外的两台机器按照上面的步骤（第一步到第八步）操作即可。

第九步：配置

```
# sed -i "s/# cluster-enabled yes/cluster-enabled yes/g;s/bind 127.0.0.1/bind 192.168.43.120
27.0.0.1/g" /data/redis/*.conf
```

注意：在另外两台机器同样执行那条命令，只需将192.168.43.120换成本机的IP即可；

第十步：重启Redis使配置生效

```
# /etc/init.d/redis_6379 restart
```

第十一步：检查

redis-cluster-01：

```
# netstat -lntp
Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address Foreign Addre
s State PID/Program name
tcp 0 0 127.0.0.1:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.120:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:6380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.120:6380 0.0.0.0:* LISTEN 9495/redis-server 1
```

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 5617/sshd
tcp 0 0 127.0.0.1:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.120:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:16380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.120:16380 0.0.0.0:* LISTEN 9495/redis-server 1
```

redis-cluster-02：

```
# netstat -lntp
Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address Foreign Addre
s State PID/Program name
tcp 0 0 127.0.0.1:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.121:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:6380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.121:6380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 5617/sshd
tcp 0 0 127.0.0.1:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.121:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:16380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.121:16380 0.0.0.0:* LISTEN 9495/redis-server 1
```

redis-cluster-03：

```
# netstat -lntp
Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address Foreign Addre
s State PID/Program name
tcp 0 0 127.0.0.1:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.122:6379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:6380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.122:6380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 5617/sshd
tcp 0 0 127.0.0.1:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 192.168.43.122:16379 0.0.0.0:* LISTEN 9485/redis-server 1
tcp 0 0 127.0.0.1:16380 0.0.0.0:* LISTEN 9495/redis-server 1
tcp 0 0 192.168.43.122:16380 0.0.0.0:* LISTEN 9495/redis-server 1
```

第十二步：Redis优化

在我们启动Redis的时候，查看Redis的启动日志，官方很友好的将一些需要优化的都给我们明显的标来了，我们只需要按照那个修改即可，如有别的需求可以自行查看官方相关参数进行修改；如下即是Redis启动时的优化提示：

```
9375:M 15 May 11:14:28.574 # WARNING: The TCP backlog setting of 511 cannot be enforced
because /proc/sys/net/core/somaxconn is set to the lower value of 128.
9375:M 15 May 11:14:28.574 # Server initialized
9375:M 15 May 11:14:28.574 # WARNING overcommit_memory is set to 0! Background save
ay fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /et
/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this
o take effect.
9375:M 15 May 11:14:28.574 # WARNING you have Transparent Huge Pages (THP) support e
abled in your kernel. This will create latency and memory usage issues with Redis. To fix this is
ue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root,
nd add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be resta
ted after THP is disabled.
```

```
# echo 1024 > /proc/sys/net/core/somaxconn
# echo "vm.overcommit_memory = 1" >>/etc/sysctl.conf
# sysctl -p
# echo never > /sys/kernel/mm/transparent_hugepage/enabled
# echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" >>/etc/rc.local
```

另外两台机器同样执行上面的操作。

第十三步：安装gem

```
# yum -y install ruby ruby-devel rubygems rpm-build
# gem install -l redis-3.0.0.gem
Successfully installed redis-3.0.0
Parsing documentation for redis-3.0.0
Installing ri documentation for redis-3.0.0
1 gem installed
```

上面的gem文件我已上传到百度云网盘

链接:https://pan.baidu.com/s/1B-Zk1KAPjUrJzCpvAZaU_w
密码:1kbv

第十四步：创建集群

```
# redis-trib.rb create --replicas 1 192.168.43.120:6379 192.168.43.121:6379 192.168.43.122:63
9 192.168.43.120:6380 192.168.43.121:6380 192.168.43.122:6380
>>> Creating cluster
auses ArgumentError in the next release
warning: this causes ArgumentError in the next release
>>> Performing hash slots allocation on 6 nodes...
Using 3 masters:
192.168.43.120:6379
192.168.43.121:6379
192.168.43.122:6379
Adding replica 192.168.43.121:6380 to 192.168.43.120:6379
Adding replica 192.168.43.122:6380 to 192.168.43.121:6379
Adding replica 192.168.43.120:6380 to 192.168.43.122:6379
M: 79e82a8357f67ac623479cc76575b39c9f7b8b39 192.168.43.120:6379 slots:0-5460 (5461 slo
s) master
M: 234af44758281d646f81ee482edd23a0f6c97237 192.168.43.121:6379 slots:5461-10922 (54
2 slots) master
M: 99d4866d25028faae3b03367a39871ab642704ab 192.168.43.122:6379 slots:10923-16383 (
461 slots) master
S: 19773196edde889421ef4d124ee1b60c2a02d5d5 192.168.43.120:6380 replicates 99d4866d
5028faae3b03367a39871ab642704ab
S: 4dde88648d44c87ff646b134b0e024043eecda40 192.168.43.121:6380 replicates 79e82a835
f67ac623479cc76575b39c9f7b8b39
S: 686cf842893bdf6e79e10bba28fcc9388848f14f 192.168.43.122:6380 replicates 234af447582
1d646f81ee482edd23a0f6c97237
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 192.168.43.120:6379)
```

```
M: 79e82a8357f67ac623479cc76575b39c9f7b8b39 192.168.43.120:6379 slots:0-5460 (5461 slo
s) master 1 additional replica(s)
S: 4dde88648d44c87ff646b134b0e024043eecda40 192.168.43.121:6380 slots: (0 slots) slave r
plicates 79e82a8357f67ac623479cc76575b39c9f7b8b39
M: 234af44758281d646f81ee482edd23a0f6c97237 192.168.43.121:6379 slots:5461-10922 (54
2 slots) master 1 additional replica(s)
S: 686cf842893bdf6e79e10bba28fcc9388848f14f 192.168.43.122:6380 slots: (0 slots) slave repl
cates 234af44758281d646f81ee482edd23a0f6c97237
S: 19773196edde889421ef4d124ee1b60c2a02d5d5 192.168.43.120:6380 slots: (0 slots) slave r
plicates 99d4866d25028faae3b03367a39871ab642704ab
M: 99d4866d25028faae3b03367a39871ab642704ab 192.168.43.122:6379 slots:10923-16383 (
461 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.
```

当出现两个OK就是集群已经创建完毕。

第十五步：Redis集群设置密码（按照生产需要自行决定）

```
# redis-cli -h 192.168.43.120 -p 6379 -c --raw
192.168.43.120:6379> config set requirepass Tnobn?dju89U
OK
192.168.43.120:6379> auth Tnobn?dju89U
OK
192.168.43.120:6379> exit
# redis-cli -h 192.168.43.120 -p 6379 -c --raw
192.168.43.120:6379> config get requirepass
NOAUTH Authentication required.
192.168.43.120:6379> auth Tnobn?dju89U
OK
192.168.43.120:6379> config get requirepass
requirepass Tnobn?dju89U
192.168.43.120:6379> exit

config set requirepass gnjRg54ish
auth gnjRg54ish
config set masterauth gnjRg54ish
```

上面的配置只是临时的，一旦Redis重启就会失效；永久修改只需要在redis.conf中添加如下配置，重即可。

```
masterauth 密码
requirepass 密码
```

Redis配置文件释义：

```
Redis的配置
    daemonize：如需要在后台运行，把该项的值改为yes
    pdifile：把pid文件放在/var/run/redis.pid，可以配置到其他地址
    bind：指定redis只接收来自该IP的请求，如果不设置，那么将处理所有请求，在生产环节中最
设置该项
    port：监听端口，默认为6379
```

timeout：设置客户端连接时的超时时间，单位为秒
loglevel：等级分为4级，debug，revbose，notice和warning。生产环境下一般开启notice
logfile：配置log文件地址，默认使用标准输出，即打印在命令行终端的端口上
database：设置数据库的个数，默认使用的数据库是0
save：设置redis进行数据库镜像的频率
rdbcompression：在进行镜像备份时，是否进行压缩
dbfilename：镜像备份文件的文件名
dir：数据库镜像备份的文件放置的路径
slaveof：设置该数据库为其他数据库的从数据库
masterauth：当主数据库连接需要密码验证时，在这里设定
requirepass：设置客户端连接后进行任何其他指定前需要使用的密码
maxclients：限制同时连接的客户端数
maxmemory：设置redis能够使用的最大内存
appendonly：开启appendonly模式后，redis会把每一次所接收到的写操作都追加到appendony.aof文件中，当redis重新启动时，会从该文件恢复出之前的状态
appendfsync：设置appendonly.aof文件进行同步的频率
vm_enabled：是否开启虚拟内存支持
vm_swap_file：设置虚拟内存的交换文件的路径
vm_max_momery：设置开启虚拟内存后，redis将使用的最大物理内存的大小，默认为0
vm_page_size：设置虚拟内存页的大小
vm_pages：设置交换文件的总的page数量
vm_max_thrrads：设置vm IO同时使用的线程数量