Redis cluster服务器安装配置

参考文献

Redis集群环境安装：<http://www.tuicool.com/articles/yU32uy3>

Redis集群安装：<http://blog.csdn.net/xu470438000/article/details/42971091>

安装包下载：<http://redis.io/download>

服务器： aws ec2

服务器：54.169.228.93(内网：172.31.18.52), 52.77.231.6(内网：172.31.12.206)

注：本文档中所有的linux命令均用红色字体标注，文件内容用绿色字体标注

## redis的安装

先在54.169.228.93 服务器上进行安装。Redis集群至少需要6个节点。两台服务器各装3个节点。

安装目录为：/ops/opt

1. 下载最新的redis-4.0-rc.tar.gz源文件到服务器
2. 解压缩

$ tar -xvf redis-4.0-rc.tar.gz

mv redis-4.0-rc redis4

1. 编译源文件

$ cd redis

$ make install

参数介绍：

make install命令执行完成后，会在src目录下生成多个可执行文件，分别是redis-server、redis-cli、redis-benchmark、redis-check-aof 、redis-check-dump，它们的作用如下：

redis-server：Redis服务器的daemon启动程序

redis-cli：Redis命令行操作工具。也可以用telnet根据其纯文本协议来操作

redis-benchmark：Redis性能测试工具，测试Redis在当前系统下的读写性能

redis-check-aof：数据修复

redis-check-dump：检查导出工具

## redis的配置

进入redis根目录，创建redis集群配置文件目录：

cd /ops/opt/redis4

mkdir cluster-conf

然后在cluster-conf目录下，创建redis集群的配置文件，因为一台服务器要启动三个节点，每个节点的配置很多相同的地方，将相同的配置信息放在redis-common.conf里面：

cluster-announce-ip 54.169.228.93

#cluster-announce-bus-port 17011

bind 172.31.18.52

protected-mode no

daemonize yes

tcp-backlog 511

timeout 0

tcp-keepalive 0

loglevel notice

databases 16

dir /ops/data/redis/cluster/

slave-serve-stale-data yes

slave-read-only yes

#not use default

repl-disable-tcp-nodelay yes

slave-priority 100

appendonly yes

appendfsync everysec

no-appendfsync-on-rewrite yes

auto-aof-rewrite-min-size 64mb

lua-time-limit 5000

cluster-enabled yes

cluster-node-timeout 15000

cluster-migration-barrier 1

slowlog-log-slower-than 10000

slowlog-max-len 128

notify-keyspace-events ""

hash-max-ziplist-entries 512

hash-max-ziplist-value 64

list-max-ziplist-entries 512

list-max-ziplist-value 64

set-max-intset-entries 512

zset-max-ziplist-entries 128

zset-max-ziplist-value 64

activerehashing yes

client-output-buffer-limit normal 0 0 0

client-output-buffer-limit slave 256mb 64mb 60

client-output-buffer-limit pubsub 32mb 8mb 60

hz 10

aof-rewrite-incremental-fsync yes

1. 参数介绍：

cluster-announce-ip:当服务器的外网IP是通过NAT映射的时候，需要配置所在服务器对外访问的IP地址。否则该参数可以不用配置

bind: 监听的IP地址

daemonize：是否以后台daemon方式运行

pidfile：pid文件位置

port：监听的端口号

timeout：请求超时时间

loglevel：log信息级别

logfile：log文件位置

databases：开启数据库的数量

save \* \*：保存快照的频率，第一个\*表示多长时间，第三个\*表示执行多少次写操作。在一定时间内执行一定数量的写操作时，自动保存快照。可设置多个条件。

rdbcompression：是否使用压缩

dbfilename：数据快照文件名（只是文件名，不包括目录）

dir：数据快照的保存目录（这个是目录）

appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数据抗风险能力，但影响效率。

appendfsync：appendonlylog如何同步到磁盘（三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步）

其余的相关配置可以参数官方网档

1. 编辑redis-17001.conf ,作为端口17001的配置。

vi redis-17001.conf

include /ops/opt/redis4/cluster-conf/redis-common.conf

port 17001

cluster-announce-port 17001

#cluster-announce-bus-port 17011

logfile "/ops/log/redis/cluster/17001.log"

maxmemory 1000000000

# volatile-lru -> remove the key with an expire set using an LRU algorithm

# allkeys-lru -> remove any key accordingly to the LRU algorithm

# volatile-random -> remove a random key with an expire set

# allkeys-random -> remove a random key, any key

# volatile-ttl -> remove the key with the nearest expire time (minor TTL)

# noeviction -> don't expire at all, just return an error on write operations

maxmemory-policy allkeys-lru

appendfilename "appendonly-17001.aof"

dbfilename "dump-17001.rdb"

cluster-config-file "nodes-17001.conf"

~

1. 因为我们需要在每台机器上部署三个节点，所以将redis-17001.conf 复制两份:

cp redis-17001.conf redis-17002.conf

cp redis-17001.conf redis-17003.conf

1. 编辑 redis-17002.conf, redis-17003.conf,

将这两个个文件中的17001全部替换成对应的端口：

vi redis-17002.conf

:%s/17001/17002/g

vi redis-17003.conf

:%s/17001/17003/g

## 启动redis

1. 进入redis安装目录的src目录，分别启动三个节点

cd src

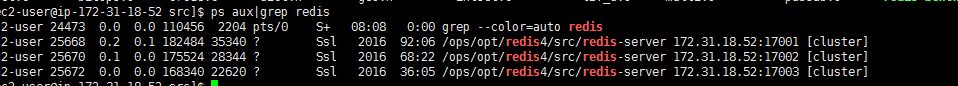
nohup ./redis-server ../cluster-conf/redis-17001.conf &

nohup ./redis-server ../cluster-conf/redis-17002.conf &

nohup ./redis-server ../cluster-conf/redis-17003.conf &

然后再检查是否启动成功:

ps aux|grep redis



现在redis三个节点都已经启动了。

1. 然后将redis4这个目录直接打包放到52.77.231.6机器上。

修改redis-common.conf :

cluster-announce-ip 52.77.231.6

#cluster-announce-bus-port 17011

bind 172.31.12.206

然后再分别按前面说明步骤分别启动三个节点并验证。

## 初始化redis集群

1. 初始化redis集群需要先安装redis集群所需环境（只需要在其中任意一台服务器上安装即可），下面是redis集群环境的安装说明，这一部份的说明来自网上，跟实际安装版本有点出入。实际安装时可根据所下载安装包路径到对应网址上找到最新的安装包下载即可。

①下载相关程序

# wget http://nchc.dl.sourceforge.net/project/libpng/zlib/1.2.8/zlib-1.2.8.tar.gz# wget https://cache.ruby-lang.org/pub/ruby/2.2/ruby-2.2.4.tar.bz2# wget https://rubygems.global.ssl.fastly.net/rubygems/rubygems-2.5.1.tgz# wget https://rubygems.global.ssl.fastly.net/gems/redis-3.2.2.gem

②编译zlib

# gunzip zlib-1.2.8.tar.gz# tar -xvf zlib-1.2.8.tar# cd zlib-1.2.8# ./configure# make# make install

③编译ruby

# bunzip2 **ruby**-2.2.4.tar.bz2

# tar -xvf **ruby**-2.2.4.tar

# **cd** **ruby**-2.2.4

# ./configure -prefix=/usr/local/**ruby**

# **make**

Generating RDoc documentation

Parsing sources...100% [967/967] vsnprintf.**c**

Generating RI format into /root/**ruby**-2.2.4/.ext/rdoc...

Files: 967

Classes: 1411 ( 581 undocumented)

Modules: 280 ( 108 undocumented)

Constants: 2160 ( 594 undocumented)

Attributes: 1156 ( 255 undocumented)

Methods: 10488 (2187 undocumented)

Total: 15495 (3725 undocumented)

75.96% documented

Elapsed: 744.0s

# **make** install

# **cp** **ruby** /bin

④编译rubygems

# tar -zxvf rubygems-2.5.1.tgz# cd rubygems-2.5.1# ruby setup.rb# cp bin/gem /bin

注:编译中出现in 'require': cannot load such file -- json/pure (LoadError)的解决方法:

# **gem** **install** **json\_pureFetching**: **json\_pure-1**.8.3.gem (100%)**Successfully** **installed** **json\_pure-1**.8.3**Parsing** **documentation** **for** **json\_pure-1**.8.3**Installing** **ri** **documentation** **for** **json\_pure-1**.8.3

1 **gem** **installed**

⑤安装gem-redis

# **gem** **install** **-l** **redis-3**.2.2.gem **Successfully** **installed** **redis-3**.2.2**Parsing** **documentation** **for** **redis-3**.2.2**Installing** **ri** **documentation** **for** **redis-3**.2.2

1 **gem** **installed**

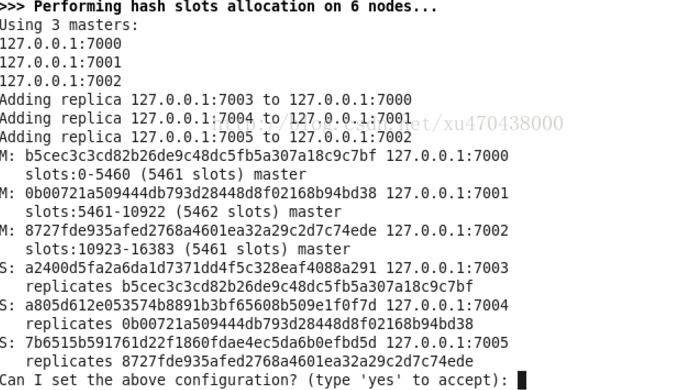
1. 现在6个redis节点都已经启动了，各个节点还没有关联。还需要通过命令来创建集群，将各个节点关联成一个集群

在其中任意一台服务器上执行创建集群的命令：

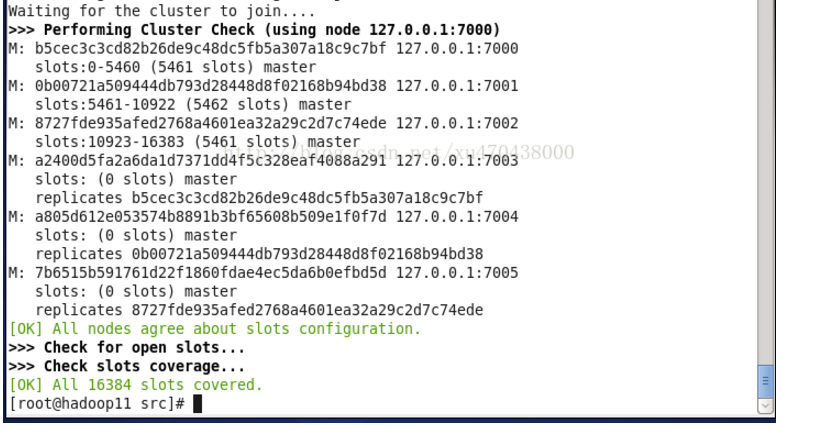
cd /ops/opt/redis4/src

./redis-trib.rb create --replicas 1 172.31.18.52:17001 172.31.18.52:17002 172.31.18.52:17003 172.31.12.206:17001 172.31.12.206:17002 172.31.12.206:17003

该命令需要启动较长时间。 当看到出现如下图所示命令：



这时输入yes ,让集群继续初始化。



因为编写该文档时，redis集群已经搭建好，所以上面两张图片均来源于网上，实际操作的时候服务器信息会是172.31.18.52和 172.31.12.206。至此集群创建成功了。

1. 启用客户端检查各节点的运行是否正常，执行相应的命令