判断主机是否安装redis中间件实施方案

# 一、方案实现架构

# 二、方案实现过程

## 2.1 未找到中间件

* **确认方法**
* 命令：ps -ef |grep redis-server |grep -v grep
* 判断标准：没有结果

## 2.1 单机模式

* **确认方法**
* 命令：ps -ef |grep redis-server |grep -v grep
* 判断标准：只有一个进程，且不带有[cluster]
* 版本：redis-server -v|awk '{print $3}'|awk -Fv= '{print $2}'
* IP：

1. 获取配置文件

* 命令：for i in `ls /|grep -Ev "bin|boot|dev|lib|lib64|media|proc|run|sys"` ;do for j in `find ${i} -type f -name "\*.conf"` ;do grep "cluster-enabled" ${j} > /dev/null && echo $j;done;done

获得配置文件并保留。

1. 过滤配置文件中IP字段

* 命令：grep -v "^#" 获得的配置文件路径 |grep bind |awk '{print $2}'

1. 获取配置文件中端口字段

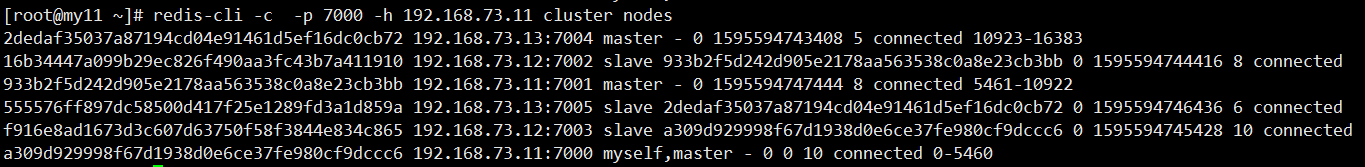
* 命令：grep -v "^#" 获得的配置文件路径|grep port |awk ‘{print $2}’

## 2.3 集群模式

* **确认方法：**
* 命令：ps –ef |grep redis-server |grep –v grep
* 判断标准：存在多个进程，且带有[cluster]
* 版本：redis-server –v|awk ‘{print $3}’|awk –Fv= ‘{print $2}’
* IP：
* **获取集群信息**

1. 确认是否存在命令 redis-cli

* 命令：which redis-cli
* 判断标准：是否输出命令路径
  1. 存在，即可使用命令获取集群信息
* 命令：redis-cli -c -p 端口号 –h 接口地址 cluster nodes



* 1. 不存在，需要查找redis源码包路径使用
* 命令：

# 三、附录

### 登陆主机执行命令

ps -ef|grep redis 判断redis是否存在，

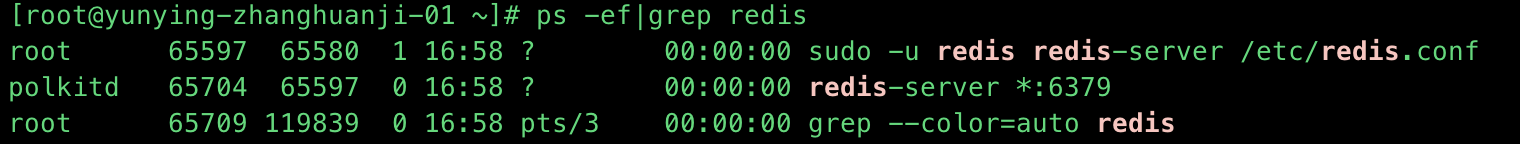
1. 未安装redis



root 27436 11036 0 16:55 pts/0 00:00:00 grep --color=auto redis

1. 安装的是单机

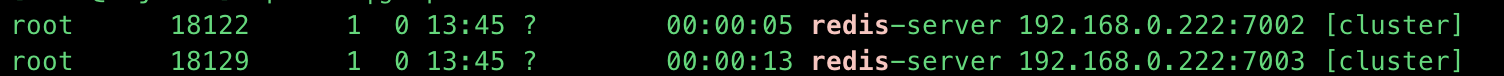
只有一个redis进程，端口6379，单机版本



polkitd 65704 65597 0 16:58 ? 00:00:00 redis-server \*:6379

1. 安装的是集群

带【cluster】说明配置开启集群模式。

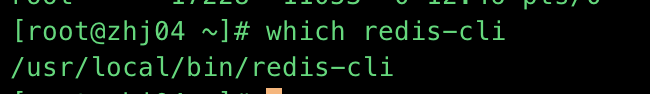


root 18122 1 0 13:45 ? 00:00:05 redis-server 192.168.0.222:7002 [cluster]

root 18129 1 0 13:45 ? 00:00:13 redis-server 192.168.0.222:7003 [cluster]

查询集群相关信息。

执行which redis-cli查询命令是否存在



which redis-cli

/usr/local/bin/redis-cli

存在则只需执行redis-cli -h IP -p 端口 -c 获取主从相关信息

IP端口如何获取参考如下

[root@zhj04 ~]# redis-cli -h 192.168.0.221 -p 7000 cluster nodes

c1d0a966f9dec0ad2c8028e544d2075260c35ff6 192.168.0.222:7003 slave 4bc6cff85da1137b8dd24f901c002bbf75733824 0 1595572415536 4 connected

3660958830a9b002e34d4f4ef0c097ef87c36da9 192.168.0.222:7002 master - 0 1595572417538 3 connected 5461-10922

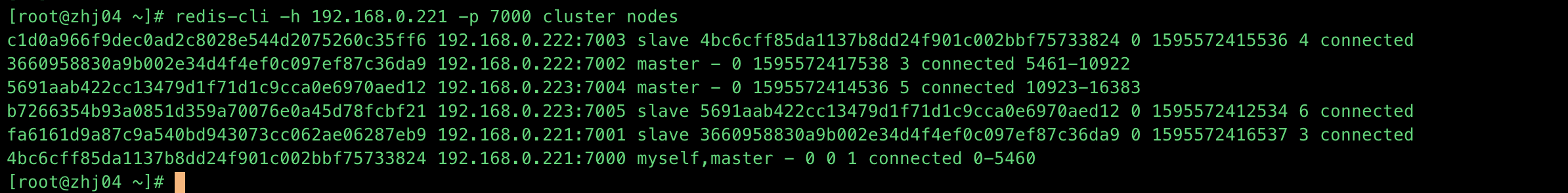
5691aab422cc13479d1f71d1c9cca0e6970aed12 192.168.0.223:7004 master - 0 1595572414536 5 connected 10923-16383

b7266354b93a0851d359a70076e0a45d78fcbf21 192.168.0.223:7005 slave 5691aab422cc13479d1f71d1c9cca0e6970aed12 0 1595572412534 6 connected

fa6161d9a87c9a540bd943073cc062ae06287eb9 192.168.0.221:7001 slave 3660958830a9b002e34d4f4ef0c097ef87c36da9 0 1595572416537 3 connected

4bc6cff85da1137b8dd24f901c002bbf75733824 192.168.0.221:7000 myself,master - 0 0 1 connected 0-5460

[root@zhj04 ~]#



master 主

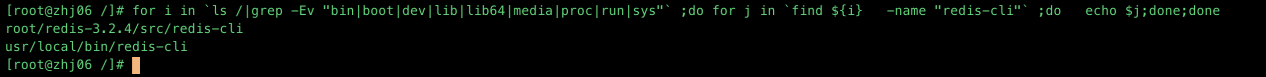
slave 从

不存在则进行搜索命令文件

[root@zhj06 /]# for i in `ls /|grep -Ev "bin|boot|dev|lib|lib64|media|proc|run|sys"` ;do for j in `find ${i} -name "redis-cli"` ;do echo $j;done;done

root/redis-3.2.4/src/redis-cli

usr/local/bin/redis-cli



搜到结果进行执行，方法同上。

1. 版本获取方法

redis-server --version

Redis server v=3.2.4 sha=00000000:0 malloc=jemalloc-4.0.3 bits=64 build=7d944fae64f5756c

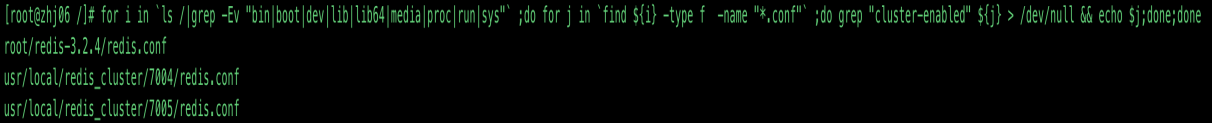
1. IP获取方法

[root@zhj06 /]# for i in `ls /|grep -Ev "bin|boot|dev|lib|lib64|media|proc|run|sys"` ;do for j in `find ${i} -type f -name "\*.conf"` ;do grep "cluster-enabled" ${j} > /dev/null && echo $j;done;done

root/redis-3.2.4/redis.conf

usr/local/redis\_cluster/7004/redis.conf

usr/local/redis\_cluster/7005/redis.conf



grep bind usr/local/redis\_cluster/7004/redis.conf

# By default, if no "bind" configuration directive is specified, Redis listens

# the "bind" configuration directive, followed by one or more IP addresses.

# bind 192.168.1.100 10.0.0.1

# bind 127.0.0.1 ::1

# internet, binding to all the interfaces is dangerous and will expose the

# following bind directive, that will force Redis to listen only into

bind 192.168.0.223 --该信息为ip地址

# 1) The server is not binding explicitly to a set of addresses using the

# "bind" directive.

# are explicitly listed using the "bind" directive.

grep port usr/local/redis\_cluster/7004/redis.conf

# Accept connections on the specified port, default is 6379 (IANA #815344).

# If port 0 is specified Redis will not listen on a TCP socket.

port 7004 --该信息为端口

# warning (only very important / critical messages are logged)

# slaveof <masterip> <masterport>

# Still a read only slave exports by default all the administrative commands

# This is important since once the transfer starts, it is not possible to serve

# It is important to make sure that this value is greater than the value

# resync is enough, just passing the portion of data the slave missed while

# A Redis master is able to list the address and port of the attached

# The listed IP and address normally reported by a slave is obtained

# Port: The port is communicated by the slave during the replication

# handshake, and is normally the port that the slave is using to

# However when port forwarding or Network Address Translation (NAT) is

# used, the slave may be actually reachable via different IP and port

# report to its master a specific set of IP and port, so that both INFO

# and ROLE will report those values.

# the port or the IP address.

# slave-announce-port 1234

# Redis supports three different modes:

# cluster node enable the cluster support uncommenting the following:

# of the failover a delay proportional to their rank.

# (However they'll always try to apply a delay proportional to their

# print graphs and obtain reports.