redis3.0集群安装

Redis 3.0 集群安装

1. 安装实例 :准备三台虚拟机 :

```
192.168.118.136
192.168.118.147
192.168.118.148
```

2. 在每个虚拟机上部署两个redis :

```
192.168.118.136:7000
192.168.118.136:7001
192.168.118.147:7000
192.168.118.147:7001
192.168.118.148:7000
192.168.118.148:7001
```

3. 环境必须

```
yum -y install gcc
yum -y install gcc-c++
yum -y install ruby
yum -y install rubygem*
yum -y install ruby-rdoc
yum -y install ruby-devel rubygems rpm-build
gem install redis // 安装redis 的第三方接口 ， 不安装貌似不行的样子
```

4 安装redis 集群

```
 # tar xvzf redis-3.0.0-beta1.tar.gz
 # mv redis-3.0.0-beta1 redis
```

编译 redis

```
 # cd ./redis/src
 # make
```

编译后 ， 将生成的文件全部移动到 bin文件下面 ， 方便使用

```
 # mkdir ../bin
 # mv mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-dump redis-cli redis-sentinel redis-server redis-trib.rb ../bin
 # cd ..
```

将 bin 下文件 分别 scp 到 各台机器上

并分别在各台机器上 复制出一份

分别修改配置文件 (此配置为最小可运行配置)

```
# vi redis.conf
 port 7000
 cluster-enabled yes    //允许集群
 cluster-config-file nodes.conf  //默认就是这个
 cluster-node-timeout 5000
 appendonly yes
```

分别运行6个redis

```
 # ./redis-server ./redis.conf &
```

在任意redis下输入命令 , 运行集群

```
 # ./redis-trib.rb create --replicas 1 192.168.118.136:7000 192.168.118.136:7001 192.168.118.147:7000 192.168.118.147:7001 192.168.118.148:7000 192.168.118.148:7001
```

5 添加新节点

```
192.168.118.147:7003
192.168.118.148:7003
```

准备好 两个redis

并启动

```
# ./redis-server redis.conf
```

添加新master 节点

输入命令redis-trib.rb add-node [需要添加节点] [原集群节点]

```
# ./redis-trib.rb add-node 192.168.118.147:7003 192.168.118.148:7000
```

添加 192.168.118.148:7003 节点作为副本

```
# ./redis-trib.rb add-node --slave 192.168.118.148:7003 192.168.118.148:7000
```

或者

```
# ./redis-trib.rb add-node --slave --master-id 4841d83e2b037c49bb078fff4b00385bf4927071 192.168.118.148:7003 192.168.118.136:7000
```

此处的 4841d83e2b037c49bb078fff4b00385bf4927071 为刚刚添加的节点 192.168.118.147:7003 的 Node id

节点ID(node id) 可在 任意redis bin目录下查看nodes.conf中有记录 还有很多方式 下面有使用到

输入命令 查看集群节点

```
# ./redis-cli -h 192.168.118.136 -p 7001
192.168.118.136:7001> cluster nodes //查看节点是否加入到集群中
192.168.118.136:7001> cluster nodes
 8cde04688d4b6df9df3d26fa4effc28643a2ee83 192.168.118.147:7001 slave eceba2187dbc166d7d8ec3554994df9f0267e960 0 1411059157653 4 connected
 ccf4a4a11cef9ad1cfe39e55d5ef5859b5c41d45 192.168.118.136:7001 myself,slave c632f8ff6ae350346dcf8a8db185720fe68ad8dd 0 0 2 connected
 f98fad39402031ec0b7e9452685d02c5336e88f6 192.168.118.148:7003 slave 694534c188782b1b0212c8a6d0959ecf8f9608a6 0 1411059158663 0 connected
 0f4f40267218d35628d57c2b7b47b5eef642de19 192.168.118.148:7001 master - 0 1411059156649 7 connected 0-5460
 ca512f21d80237fcb27f2ade91d4c1bebc6b5f5e 192.168.118.136:7000 slave 0f4f40267218d35628d57c2b7b47b5eef642de19 0 1411059158157 7 connected
 c632f8ff6ae350346dcf8a8db185720fe68ad8dd 192.168.118.148:7000 master - 0 1411059158158 5 connected 5461-10922
 eceba2187dbc166d7d8ec3554994df9f0267e960 192.168.118.147:7000 master - 0 1411059157653 3 connected 10923-16383
 694534c188782b1b0212c8a6d0959ecf8f9608a6 192.168.118.147:7003 master - 0 1411059157150 0 connected
```

可知 新添加的节点没有分配到 槽

下面开始给新节点分配 hash 槽

```
# ./redis-trib.rb reshard [任意节点地址:port]
 How many slots do you want to move (from 1 to 16384)? 4096 //由于这又4队主从节点16384/4 [按照情况分配数量]
 .......
 What is the receiving node ID? 4841d83e2b037c49bb078fff4b00385bf4927071 //注意此处的ID
 Please enter all the source node IDs.
   Type 'all' to use all the nodes as source nodes for the hash slots.
   Type 'done' once you entered all the source nodes IDs.
 Source node #1: all //也可以是某个节点的ID
  ......
 Do you want to proceed with the proposed reshard plan (yes/no)? yes //此处输入yes 执行操作,输入no 取消分配
```

分配好之后输入命令 检查 集群状态

```
# ./redis-trib.rb check [任意节点地址:port]
[root@master addm]# ./redis-trib.rb check 192.168.118.148:7001
 Connecting to node 192.168.118.136:7000: OK
 Connecting to node 192.168.118.147:7003: OK
 Connecting to node 192.168.118.148:7001: OK
 Connecting to node 192.168.118.136:7001: OK
 Connecting to node 192.168.118.148:7000: OK
 Connecting to node 192.168.118.147:7001: OK
 Connecting to node 192.168.118.147:7000: OK
 >>> Performing Cluster Check (using node 192.168.118.136:7000)
 M: 6fb72d5d6f14c67840348c0d14767f2541ae128a 192.168.118.136:7000
    slots:12288-16383 (4096 slots) master
    1 additional replica(s)
 M: 4841d83e2b037c49bb078fff4b00385bf4927071 192.168.118.147:7003
    slots:0-1364,5461-6826,10923-12287 (4096 slots) master
    0 additional replica(s)
 S: 142ef0fd250be201a4f682163245a98d58d230c3 192.168.118.148:7001
    slots: (0 slots) slave
    replicates cde6ec4b98a8e2c5d4020a5e545a7b95c533737d
 S: efce5470e3de0a2ba08211c7fb0d1a860051ba4a 192.168.118.136:7001
    slots: (0 slots) slave
    replicates 6fb72d5d6f14c67840348c0d14767f2541ae128a
 M: d11986f2ccfad296c503684ac372ece29c587f14 192.168.118.148:7000
    slots:1365-5460 (4096 slots) master
    1 additional replica(s)
 S: 0cc4699066987d99391b0faccddf25e8be44c62c 192.168.118.147:7001
    slots: (0 slots) slave
    replicates d11986f2ccfad296c503684ac372ece29c587f14
 M: cde6ec4b98a8e2c5d4020a5e545a7b95c533737d 192.168.118.147:7000
    slots:6827-10922 (4096 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.
```

结果中

```
M: 4841d83e2b037c49bb078fff4b00385bf4927071 192.168.118.147:7003
   slots:0-1364,5461-6826,10923-12287 (4096 slots) master
```

6 集群相关命令

集群

CLUSTER INFO 打印集群的信息

CLUSTER NODES 列出集群当前已知的所有节点（node），以及这些节点的相关信息。

节点

CLUSTER MEET <ip> <port> 将 ip 和 port 所指定的节点添加到集群当中，让它成为集群的一份子。

CLUSTER FORGET <node_id> 从集群中移除 node_id 指定的节点。

CLUSTER REPLICATE <node_id> 将当前节点设置为 node_id 指定的节点的从节点。

CLUSTER SAVECONFIG 将节点的配置文件保存到硬盘里面。

槽(slot)

CLUSTER ADDSLOTS <slot> [slot ...] 将一个或多个槽（slot）指派（assign）给当前节点。

CLUSTER DELSLOTS <slot> [slot ...] 移除一个或多个槽对当前节点的指派。

CLUSTER FLUSHSLOTS 移除指派给当前节点的所有槽，让当前节点变成一个没有指派任何槽的节点。

CLUSTER SETSLOT <slot> NODE <node_id> 将槽 slot 指派给 node_id 指定的节点，如果槽已经指派给另一个节点，那么先让另一个节点删除该槽>，然后再进行指派。

CLUSTER SETSLOT <slot> MIGRATING <node_id> 将本节点的槽 slot 迁移到 node_id 指定的节点中。

CLUSTER SETSLOT <slot> IMPORTING <node_id> 从 node_id 指定的节点中导入槽 slot 到本节点。

CLUSTER SETSLOT <slot> STABLE 取消对槽 slot 的导入（import）或者迁移（migrate）。

键

CLUSTER KEYSLOT <key> 计算键 key 应该被放置在哪个槽上。

CLUSTER COUNTKEYSINSLOT <slot> 返回槽 slot 目前包含的键值对数量。

CLUSTER GETKEYSINSLOT <slot> <count> 返回 count 个 slot 槽中的键。

分享：redis3 相关博文

http://blog.csdn.net/myrainblues/article/details/25881535

./bin/redis-trib.rb check 192.168.56.106:7007   查看系信息