NoSQL数据库管理

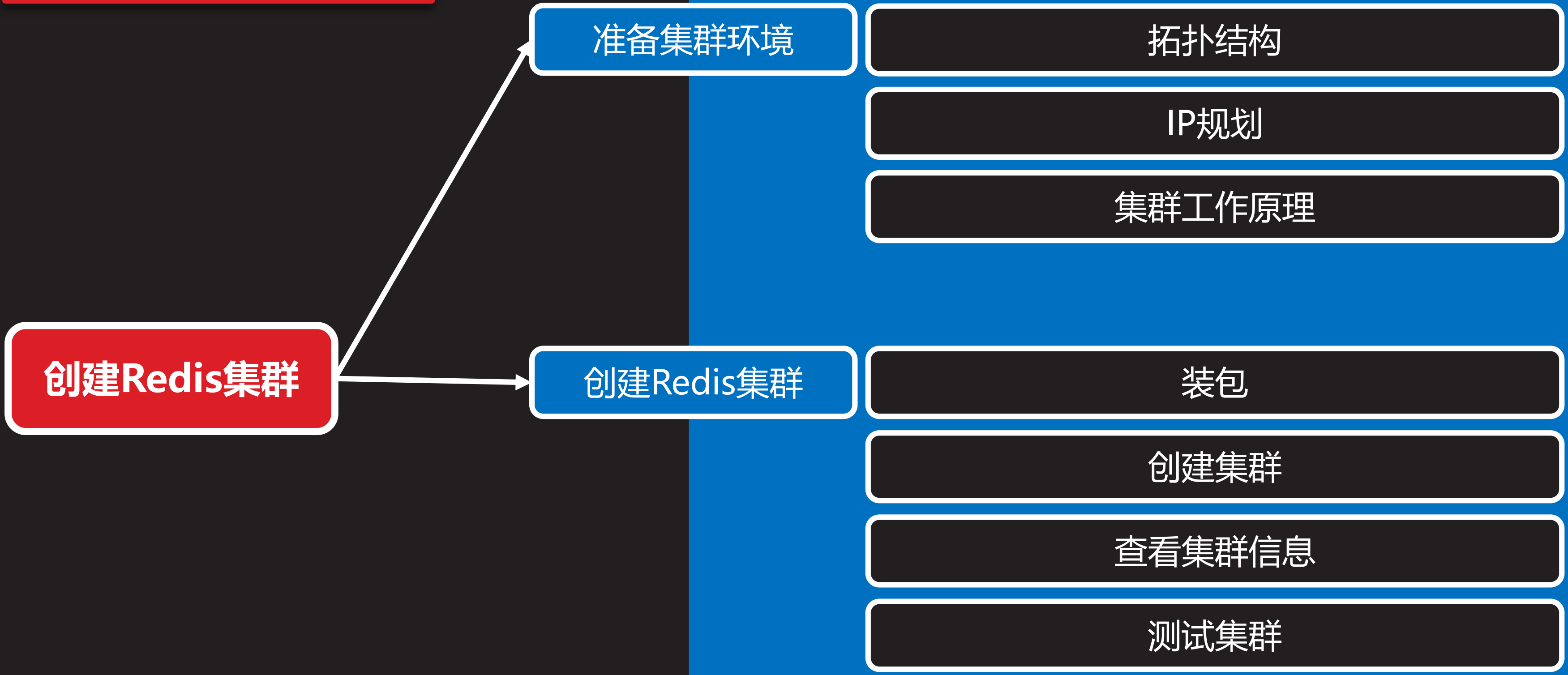
NSD　NoSQL

DAY02

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	创建Redis集群
	10:30 ~ 11:20	
	11:30 ~ 12:00	管理集群
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑

创建Redis集群



创建Redis集群

准备集群环境

- 拓扑结构
- IP规划
- 集群工作原理

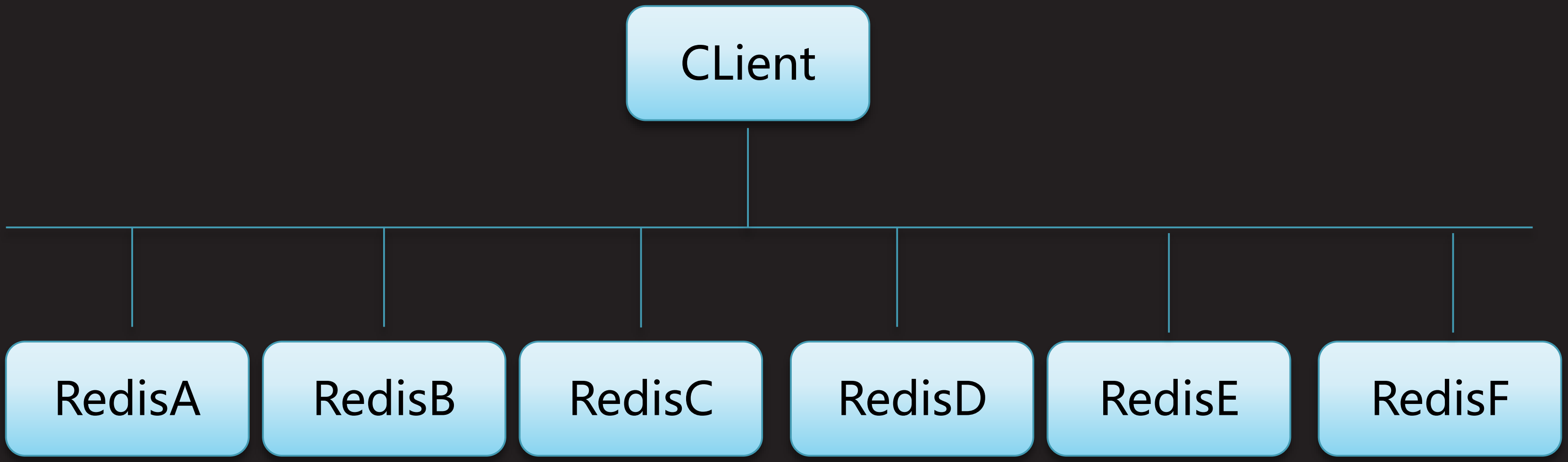
创建Redis集群

- 装包
- 创建集群
- 查看集群信息
- 测试集群

准备集群环境

拓扑结构

- 六台Redis服务器



知识讲解

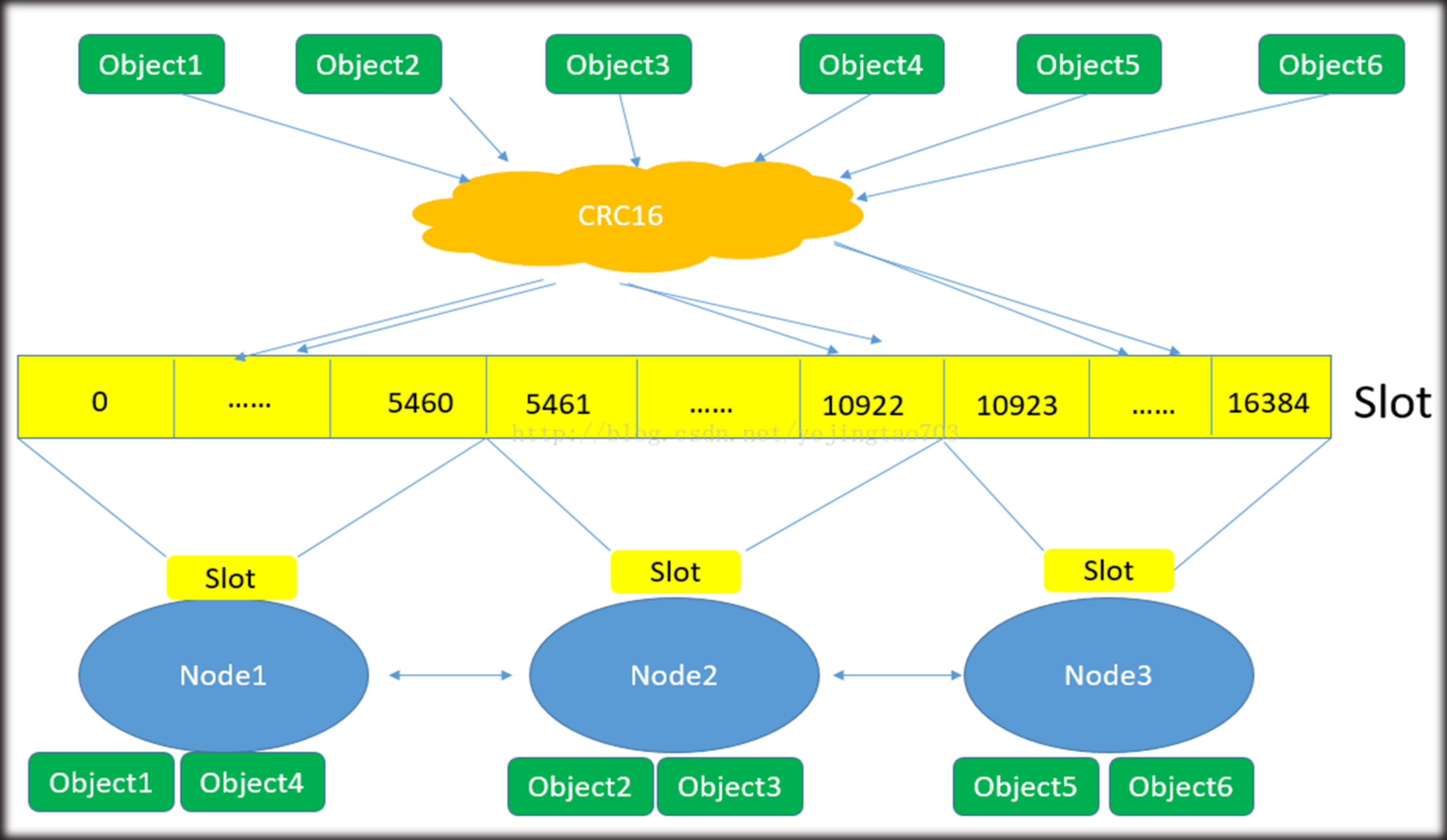
IP地址规划

知识讲解

- redis 服务器 IP 地址及端口规划

– redisA	192.168.4.51	6351
– redisB	192.168.4.52	6352
– redisC	192.168.4.53	6353
– redisD	192.168.4.54	6354
– redisE	192.168.4.55	6355
– redisF	192.168.4.56	6356

集群工作原理


Object1
Object2
Object3
Object4
Object5
Object6
CRC16
0
……
5460
5461
……
10922
10923
……
16384
Slot
Slot
Slot
Slot
Node1
Node2
Node3
Object1
Object4
Object2
Object3
Object5
Object6

创建Redis集群

装包

知识讲解

- 为6台主机安装并运行redis服务

```
# yum  -y  install  gcc  gcc-c++
# tar  -zxvf  redis-4.0.8.tar.gz
# cd  redis-4.0.8/
# make
# make  install
# ./utils/install_server.sh
```

- 调整配置文件

```
# vim  /etc/redis/redis.conf
bind  IP地址                                    //只写物理接口IP地址
daemonize    yes                                //守护进程方式运行
port  xxxx                                      //端口号不要使用默认的6379
cluster-enabled  yes                            //启用集群
cluster-config-file  nodes-xxxx.conf            //指定集群信息文件
cluster-node-timeout  5000                      //请求超时 5 秒
```

装包（续1）

- 查看服务信息

```
[root@host51 ~]# /etc/init.d/redis_6379 status
Redis is running (21201)
[root@host51 ~]#
[root@host51 ~]# netstat -utnlp  | grep redis-server

tcp        0      0 192.168.4.51:6351       0.0.0.0:*               LISTEN
21201/redis-server

tcp        0      0 192.168.4.51:16351      0.0.0.0:*               LISTEN
21201/redis-server
[root@host51 ~]#
```

集群通信端口 默认服务端口+10000

装包（续2）

- 查看集群信息

```
[root@host51 ~]# redis-cli  -h 192.168.4.51 -p 6351
192.168.4.50:6350> cluster info
cluster_state:fail
cluster_slots_assigned:0
cluster_slots_ok:0
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:1
cluster_size:0
.....
192.168.4.50:6350>
192.168.4.50:6350> cluster  nodes
e081313ec843655d9bc5a17f3bed3de1dccb1d2b
192.168.4.50:6350@16350 myself,master - 0 1530781129976 11
connected
192.168.4.50:6350>
```

创建集群

- 在选中的一台redis服务器上，执行创建集群脚本
  - 部署ruby脚本运行环境
  - 创建集群

```
# yum  -y  install   ruby   rubygems
# rpm  -ivh  --nodeps  ruby-devel-2.0.0.648-30.el7.x86_64.rpm
# gem  install  redis-3.2.1.gem

# cd  redis-3.2.0/src/
# ./redis-trib.rb   create  --replicas 1   host:port  host:port   ......
```

--replicas 1 ，自动为每一个master节点分配一个slave节点

创建集群（续1）

知识讲解

- 创建集群

```
[root@host51 ~]# redis-trib.rb create --replicas 1   \
192.168.4.51:6351 192.168.4.52:6352   \
192.168.4.53:6353 192.168.4.54:6354   \
192.168.4.55:6355 192.168.4.56:6356   \
>>> Creating cluster
>>> Performing hash slots allocation on 6 nodes...
Using 3 masters:
192.168.4.51:6351
192.168.4.52:6352
192.168.4.53:6353
......
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

查看集群信息

- 任意一台主机访问本机的redis服务查看即可

  > cluster info   //查看集群信息

  > cluster nodes   //查看集群节点信息

```
# redis-cli  -h  192.168.4.52  -p  6352
192.168.4.52:6352> cluster   info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
.. ..
cluster_known_nodes:6
cluster_size:3
.. ..
```

查看集群信息（续1）

- 任意一台主机访问本机的redis服务查看即可

```
> cluster  info                //查看集群信息
> cluster  nodes               //查看集群节点信息
```

```
# redis-cli  -h  192.168.4.52  -p  6352
192.168.4.52:6352> cluster  nodes
6e841e2610c3d4d……. 192.168.4.51:6351@16351 master ……
2de5136be52a327……. 192.168.4.54:6354@16354 slave ……
3198014263d26a6……. 192.168.4.56:6356@16356 master …….
82781de818fb83cc…… 192.168.4.53:6353@16353 master  …….
b52feba0e6422b44….. 192.168.4.52:6352@16352 myself,slave ……
a0c8e18619828487….. 192.168.4.55:6355@16355 slave ……
.. ..
```

测试集群

知识讲解

- 在客户端访问任意一台master主机存数据
  - redis-cli -c -h ip地址 -p 端口

  > set key values //存数据

```
[root@host51 ~]# redis-cli  -c -h 192.168.4.51 -p 6351
192.168.4.51:6351> set school tarena
-> Redirected to slot [8455] located at 192.168.4.52:6352
OK
192.168.4.52:6352> set class linux
OK
192.168.4.52:6352> set pay 26800
-> Redirected to slot [4013] located at 192.168.4.51:6351
OK
```

测试集群（续1）

- 在客户端访问任意一台master主机取数据
  - redis-cli -c -h ip地址 -p 端口

  > get key //取数据

```
[root@host51 ~]# redis-cli  -c -h 192.168.4.53 -p 6353
192.168.4.53:6353> get name
-> Redirected to slot [5798] located at 192.168.4.52:6352
"bob"
192.168.4.52:6352> keys *
1) "school"
2) "name"
3) "class"
4) "age"
192.168.4.52:6352> get age
"19"
```

知识讲解

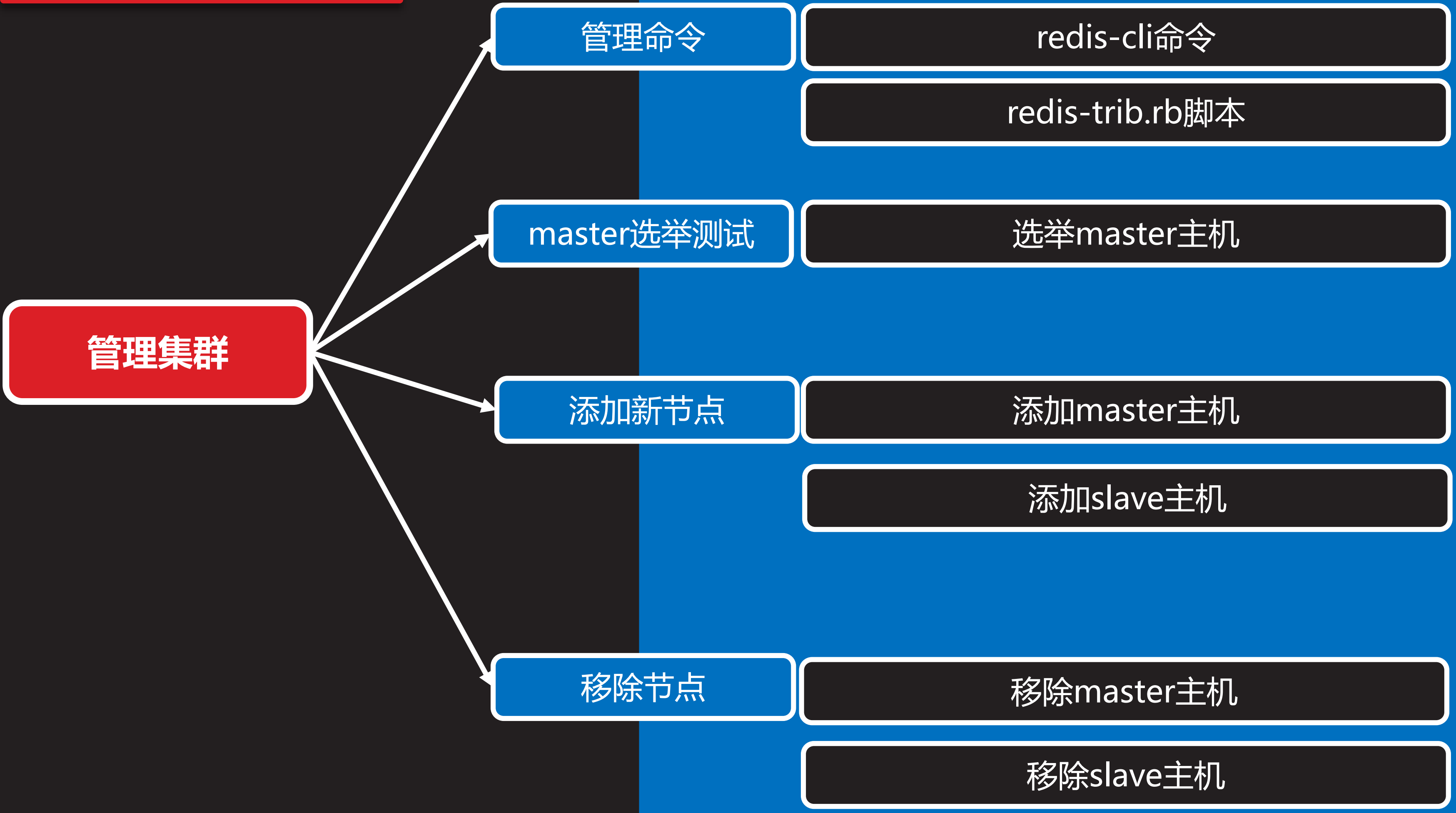
案例1：部署redis集群

课堂练习

具体要求如下：

- 准备集群环境
- 安装redis并创建集群
- 查看集群信息

管理集群



管理集群

管理命令

redis-cli命令

redis-trib.rb脚本

master选举测试

选举master主机

添加新节点

添加master主机

添加slave主机

移除节点

移除master主机

移除slave主机

管理命令

Tedu.cn 达内教育

redis-cli命令

知识讲解

- 查看命令帮助
  - redis-cli -h
- 常用选项
  - -h IP地址
  - -p 端口
  - -c 集群模式

redis-trib.rb脚本

- 语法格式
  - redis-trib.rb 选项 参数
- 选项
  - add-node 添加master主机
  - check 检测集群
  - reshard 重新分片
  - add-node --slave 添加slave主机
  - del-node 删除主机

master选举测试

选举master主机

- 停止master 主机的 Redis服务
  - master宕机后对应的slave自动被选举为master
  - 原master启动后 会自动配置为当前master的slave
- 查看集群主机信息
  - redis-cli -h master_ip   -p master_port

```
[root@host51 ~]# redis-cli  -h 192.168.4.51 -p 6351
192.168.4.51:6351> cluster  info
```

添加新节点

添加master主机

- 部署一台新redis服务器
  - 装包
  - 初始化
  - 启用集群配置
  - 重启服务
- 添加master主机步骤
  - 添加master主机
  - 检查主机
  - 重新分片

知识讲解

添加master主机（续1）

知识讲解

- 添加master主机
  - 添加时不指定主机角色，默认新主机被选为master
  - # ./redis-trib.rb add-node 新主机Ip:端口 192.168.4.51:6351

```
[root@host51 ~]# redis-trib.rb  add-node 192.168.4.50:6350
192.168.4.51:6351
>>> Adding node 192.168.4.50:6350 to cluster
192.168.4.51:6351
......
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Send CLUSTER MEET to node 192.168.4.50:6350 to
make it join the cluster.
[OK] New node added correctly.
```

添加master主机（续2）

知识讲解

- 检测集群主机
  - ./redis-trib.rb check 192.168.4.51:6351
  - 主机角色为master
  - 无槽位数量

```
[root@host51 ~]# redis-trib.rb  check 192.168.4.51:6351
>>> Performing Cluster Check (using node 192.168.4.51:6351)
……
……
M: e081313ec843655d9bc5a17f3bed3de1dccb1d2b
192.168.4.50:6350
   slots: (0 slots) master
   0 additional replica(s)
……
……
```

添加master主机（续3）

- 重新分片
  - ./redis-trib.rb reshard   192.168.4.51:6351
  - 指定移出hast槽个数
  - 指定接收hash槽主机ID
  - 指定移出hash槽主机ID

```
[root@host51 ~]# redis-trib.rb  reshard 192.168.4.51:6351
How many slots do you want to move (from 1 to 16384)? 4096
What is the receiving node ID?
e081313ec843655d9bc5a17f3bed3de1dccb1d2b
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 #1:3550af084c94889ae2d01103e6da5793fdd851fc
Source node #2:done
……
Do you want to proceed with the proposed reshard plan (yes/no)? yes
```

知识讲解

添加slave主机

- 部署一台新redis服务器
  - 装包
  - 初始化
  - 启用集群配置
  - 运行服务
- 添加slave主机

```
# ./redis-trib.rb  add-node  --slave  [ --master-id  id值  ]
   ip地址:端口  192.168.4.51:6351
```

如果不指定主节点的id 的话，会把新节点 随机添加为 从节点 最少的主的从

添加slave主机（续1）

- 添加slave主机

```
[root@host51 ~]# redis-trib.rb  add-node  --slave
192.168.4.57:6357  192.168.4.51:6351
>>> Adding node 192.168.4.57:6357 to cluster
192.168.4.51:6351
>>> Performing Cluster Check (using node 192.168.4.51:6351)
......
......
[OK] All 16384 slots covered.
Automatically selected master 192.168.4.51:6351
>>> Send CLUSTER MEET to node 192.168.4.57:6357 to make
it join the cluster.
Waiting for the cluster to join.
>>> Configure node as replica of 192.168.4.51:6351.
[OK] New node added correctly.
```

移除节点

移除master主机

知识讲解

- 配置步骤
  - 重新分片释放占用的hash槽
  - 移除master主机

```
# redis-trib.rb reshard 192.168.4.51:6351

# redis-trib.rb del-node  192.168.4.51:6351   maste主机id值
```

移除master主机（续1）

- 重新分片释放占用的hash槽
  - 指定移出hash槽个数
  - 指定接收hash槽主机ID
  - 指定移出hash槽主机ID

知识讲解

```
[root@host51 ~]# redis-trib.rb reshard 192.168.4.51:6351
How many slots do you want to move (from 1 to 16384)? 4096
What is the receiving node ID?
4361720c3978aa02347076218580a103c60a6d7f
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:e081313ec843655d9bc5a17f3bed3de1dccb1d2b
Source node #2:done
......
Do you want to proceed with the proposed reshard plan (yes/no)? yes
```

移除master主机（续2）

- 移除master主机

```
# redis-trib.rb del-node  192.168.4.51:6351   maste主机id值

[root@host51 ~]# redis-trib.rb del-node 192.168.4.51:6351  \
e081313ec843655d9bc5a17f3bed3de1dccb1d2b
>>> Removing node
e081313ec843655d9bc5a17f3bed3de1dccb1d2b from cluster
192.168.4.51:6351
>>> Sending CLUSTER FORGET messages to the cluster...
>>> SHUTDOWN the node.
[root@host51 ~]#
```

知识讲解

移除slave主机

- 移除slave主机
  - 从节点主机没有槽位范围，直接移除即可
  - redis-trib.rb del-node 192.168.4.51:6351 主机id值

```
# redis-trib.rb  del-node 192.168.4.51:6351  \
f6649ea99b2f01faca26217691222c17a3854381
>>> Removing node f6649ea99b2f01faca26217691222c17a3854381
      from cluster 192.168.4.51:6351
>>> Sending CLUSTER FORGET messages to the cluster...
>>> SHUTDOWN the node.
```

案例2：管理redis集群

具体要求如下：

- 练习添加主机
- 练习删除主机

课堂练习

总结和答疑

总结和答疑

管理集群

redis-trib.rb脚本选项总结

管理集群

redis-trib.rb脚本选项总结

- redis-trib.rb 常用选项

选项	作用
create	创建集群
check	检查集群
reshard	重新分片
del-node	删除主机
add-node --slave	添加slave主机
add-node	添加master主机