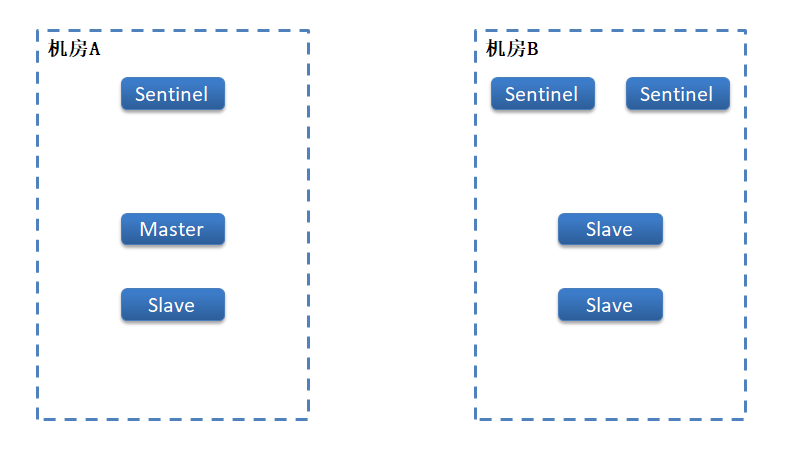
# 一、方案描述：

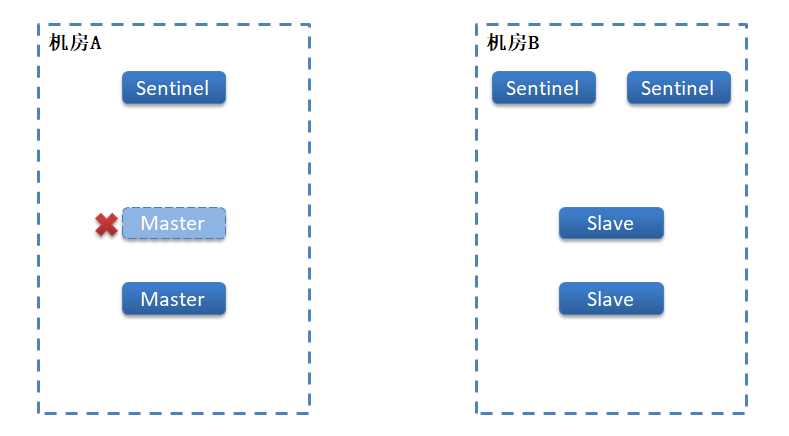
双中心组合搭建REDIS集群采用一主三从三哨兵架构：



* + 建议至少三个节点，一个sentinel要接管failover过程必须得到至少半数以上的sentinel认可，三个sentinel节点挂掉一个集群仍然正常工作；
  + 为了确保机房故障（断电或者网络分区）下仍然能够进行主从切换，则主节点所在的机房sentinel个数应该小于集群的一半，这样当主节点所在的机房故障时，另一个机房中存活有超过半数的sentinel，从而接管failover过程（在另一个机房的slave节点中提升一个当做主节点）。

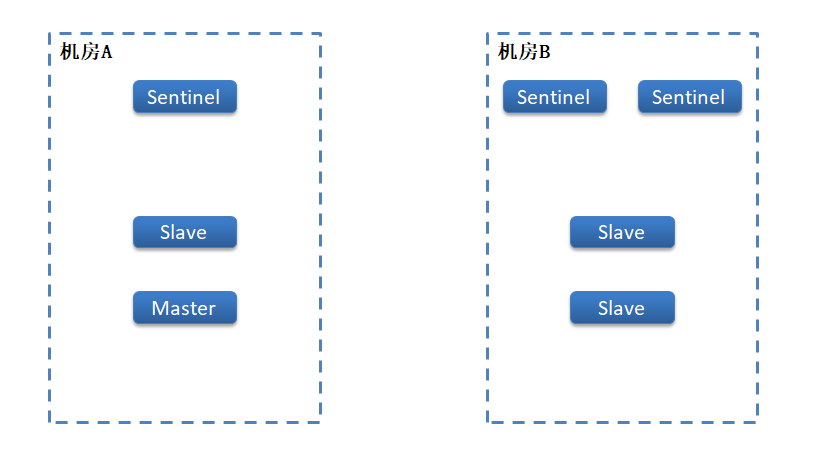
故障切换

* 当主节点挂后，哨兵察觉到Master挂掉后，会启动故障自动切换，选出一个slave作为新的主节点，这里尽量优先选择同机房的slave做主从切换（脚本动态切换slave节点的优先级配置，另一种就是在redis节点相互请求信息中带上DC-id作为机房标识），仅当机房故障时才进行跨机房故障转移；



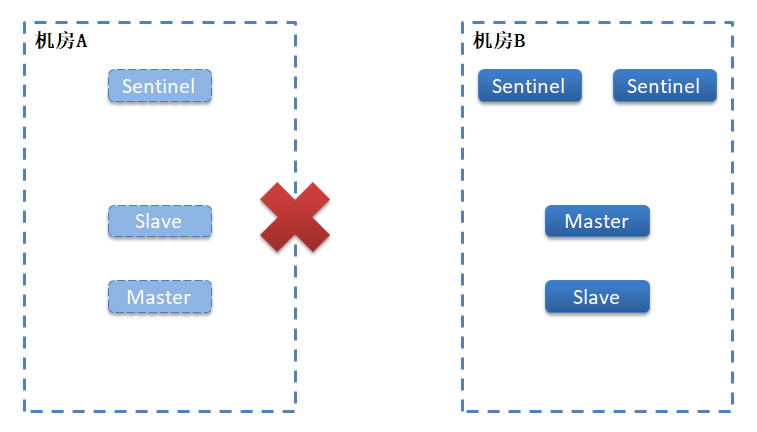
主节点切换

* 当故障的主节点再次启动加入主从模式中，它将作为新主节点的从节点，同步当前主节点的数据。



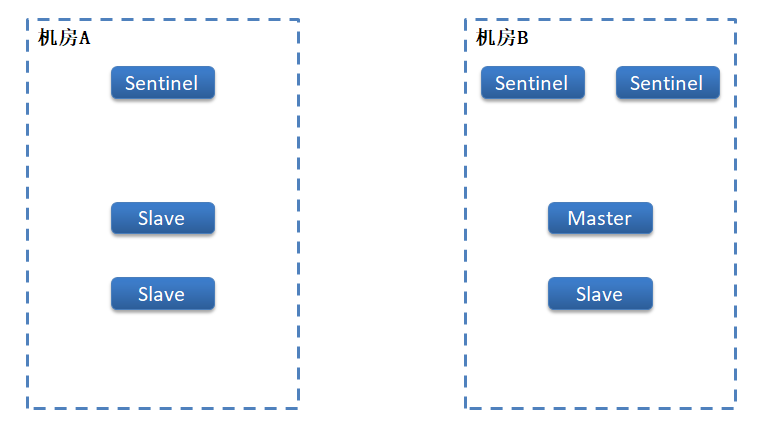
主变从

* 接下来讨论机房故障的情况。
* 机房B故障：A的sentinel发现其他Sentinel和B的Slave节点都失效，此时虽然哨兵的failover功能无法使用，但是A的Master和Slave节点仍然能够正常读写工作，此时只要等待B机房恢复即可，恢复重启后sentinel重新监测A机房master节点，slave节点会从master节点同步数据（redis自身确保，可能是全量同步也可能是增量同步）；
* 机房A故障：B的哨兵发起failover过程，将其中的一个slave节点提升为master节点，此时B机房可以正常提供读写操作。



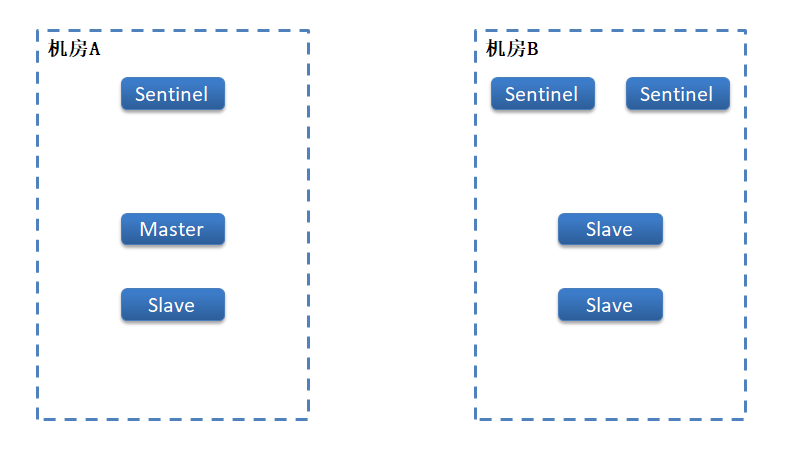
机房A故障，B接管

* 此时机房A恢复，sentinel更新自己和redis节点的配置将B的master节点确认为新的主节点（这个是由sentinel自身机制确保，B中的sentinel配置版本号较新，A会以B为准），A机房的redis节点变为slave并从B的master节点进行数据同步；



机房A恢复后，最新的主从结构

* 到这里redis主从切换都没有问题，单节点的故障也不会影响整个主从结构。
* 现在的问题是Master节点如何回到主机房。首先需要到机房B找到主节点，杀掉机房B的主节点，此时主节点会自动回到机房A。存在的问题是当机房A与机房B失去通信期间，此时机房A存储的数据与机房B存储的数据是不一致的，当网络恢复后，机房B的数据会覆盖机房A的数据。



机房A恢复后，主节点重新回到主机房

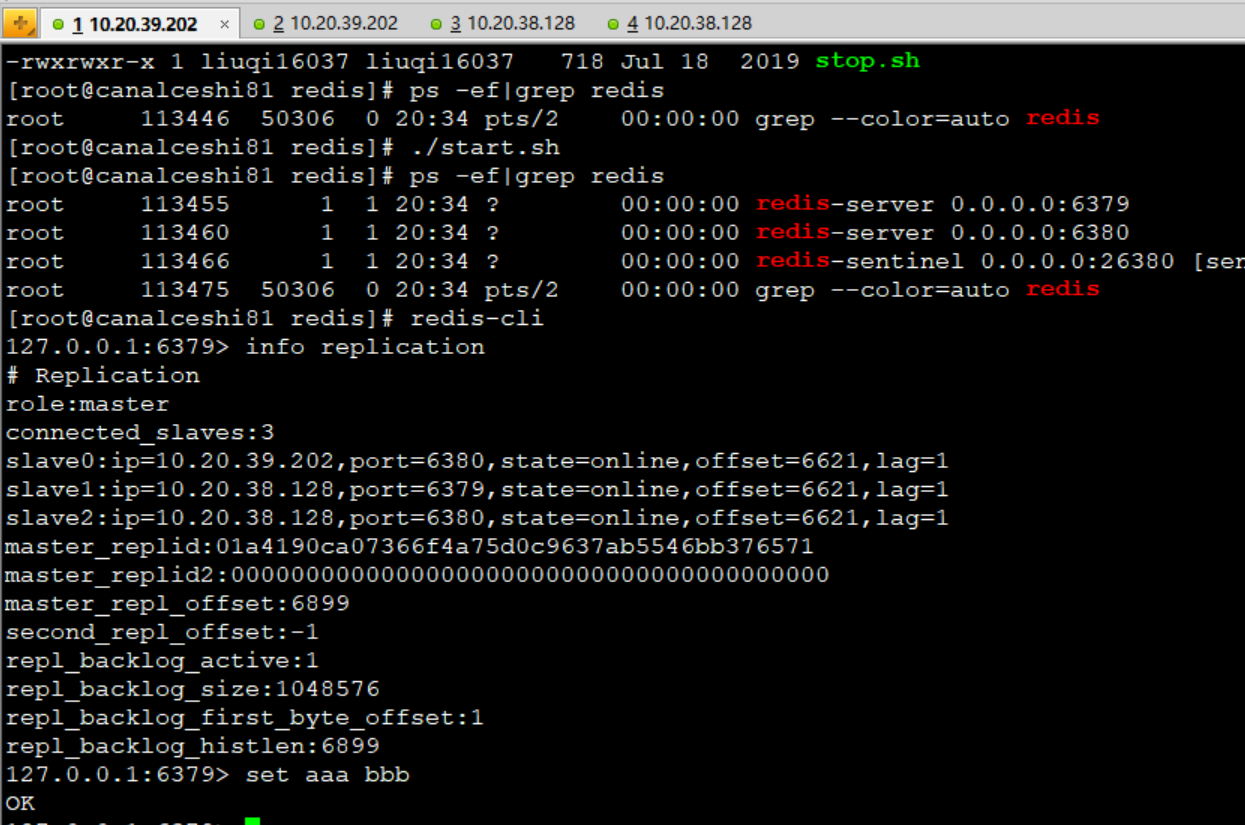
# 二、机器准备及验证：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 机器ip | 作用 | 端口 |  |
|  | 10.20.39.202 | 数据节点 | 36381 | 主机房机器02 |
| 哨兵节点 | 36380 |
| 数据节点 | 46381 | 主机房机器03 |
| 备机房 | 192.168.73.36 | 数据节点 | 36381 | *备机房机器01* |
| 哨兵节点 | 36380 |
| 数据节点 | 46381 | *备*机房机器02 |
| 哨兵节点 | 46380 |

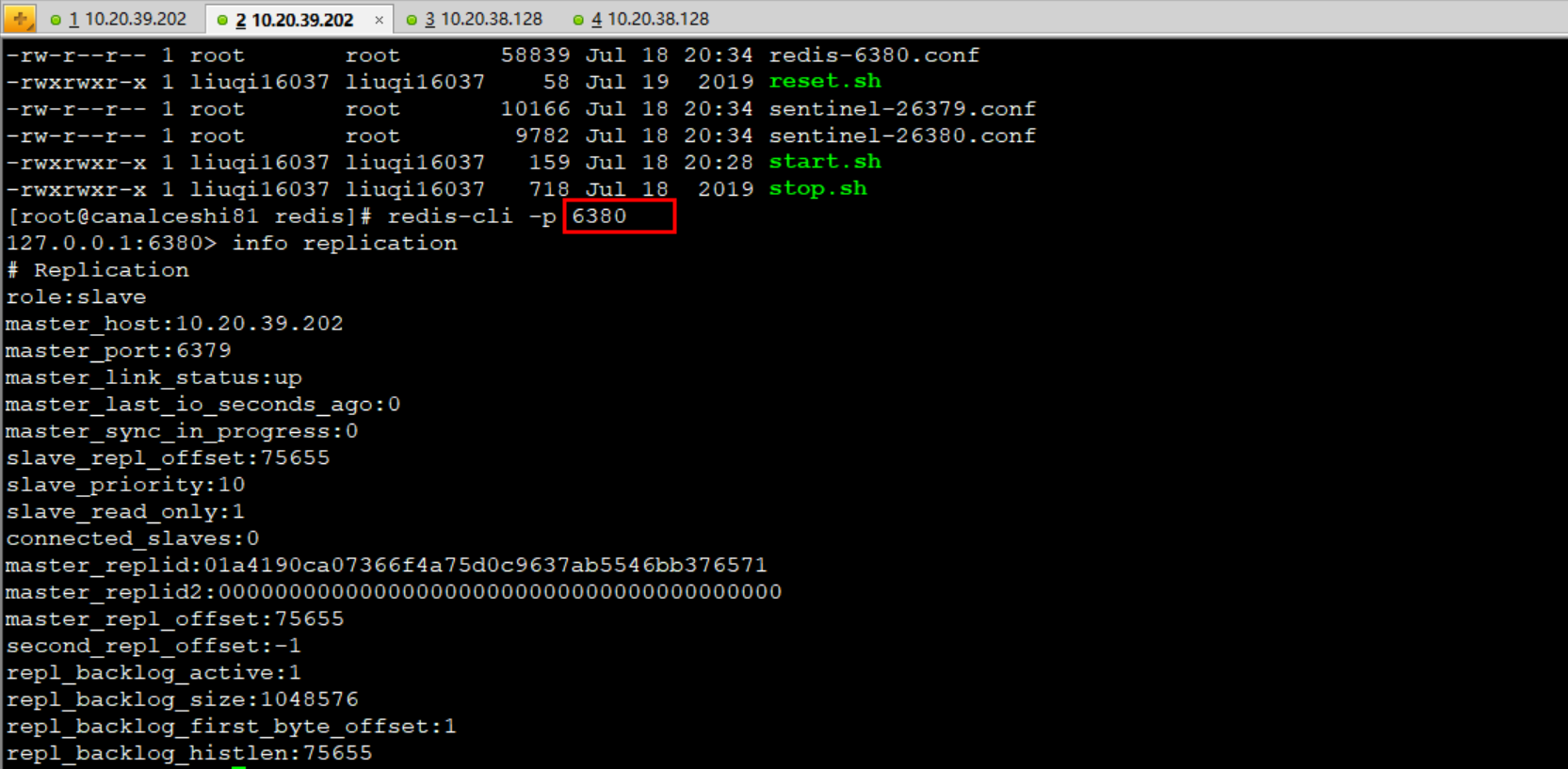
# 三、高可用验证方案1：模拟机器宕机

## 1、开始启动4个节点，初始条件下：

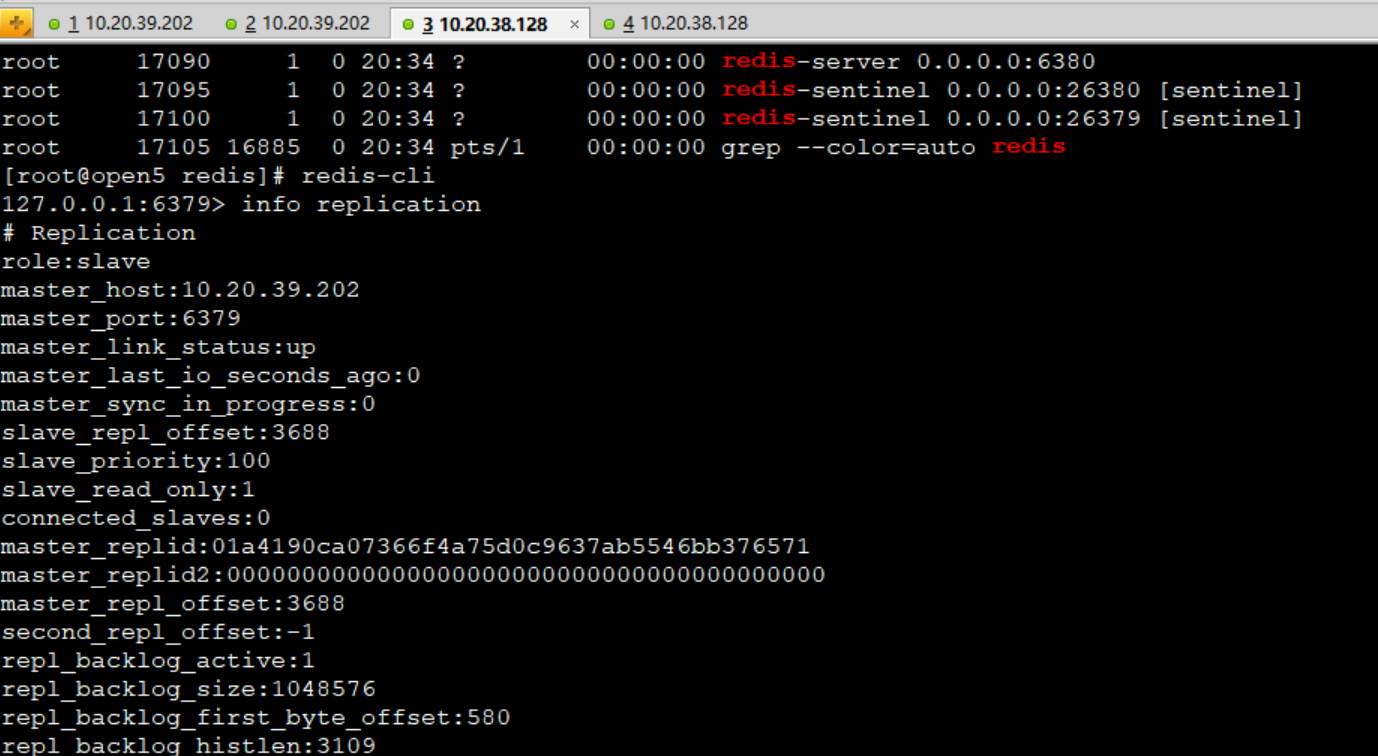
### （1）主机房机器01（主节点）截图：



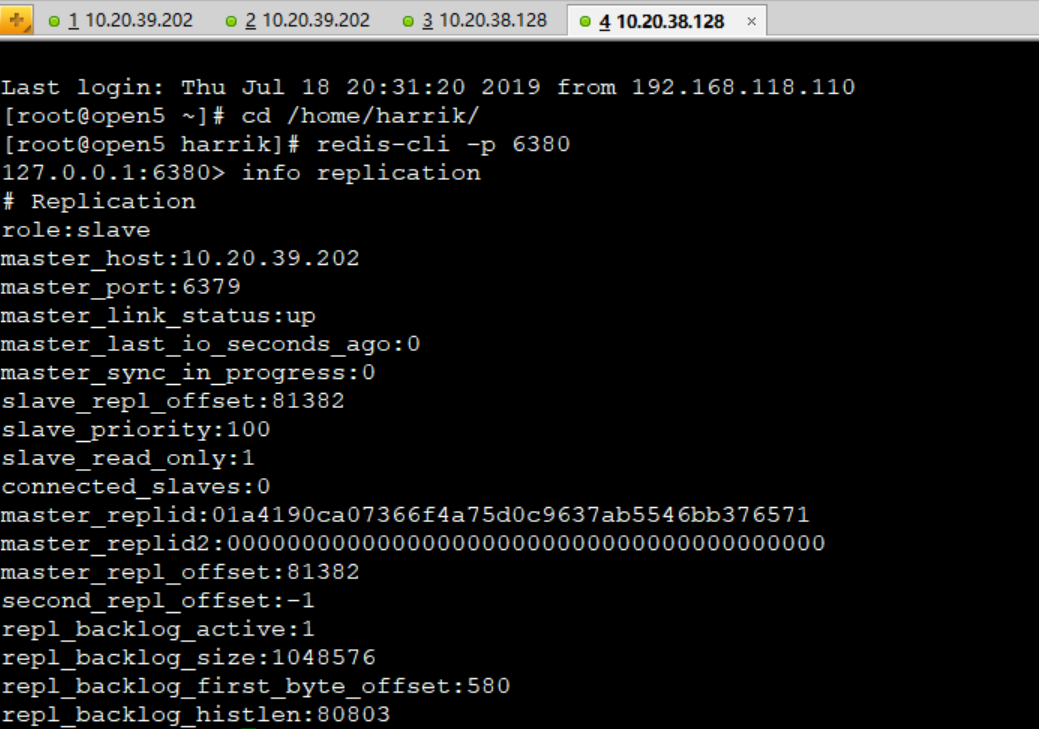
### （2）主机房机器02（从节点）截图：



### （3）备机房机器01（从节点）截图：

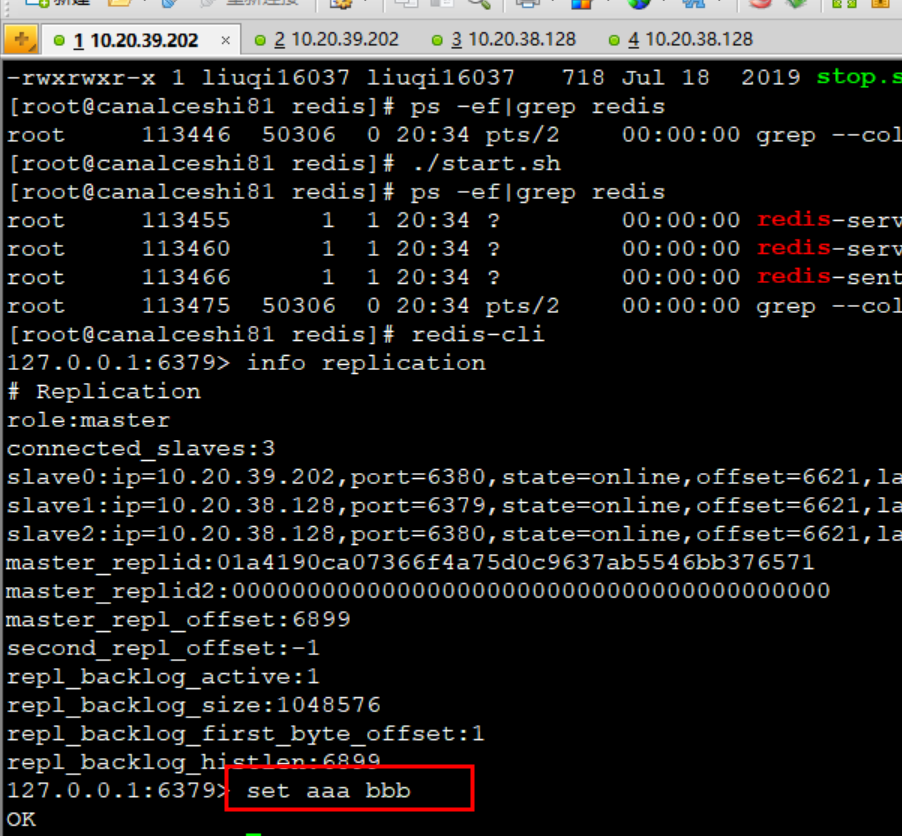


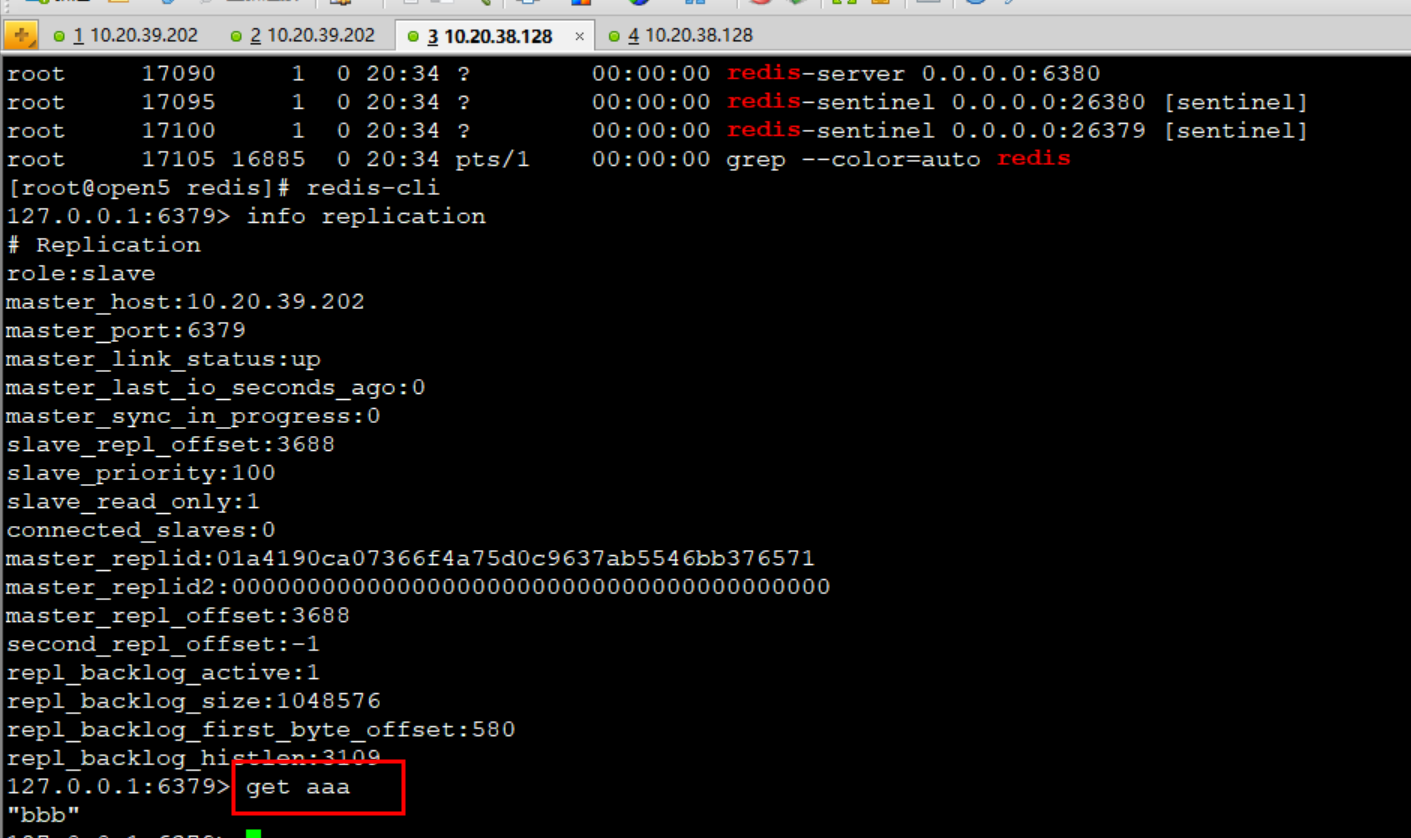
### （4）备机房机器02（从节点）截图：



## 2、校验主从机房节点数据同步

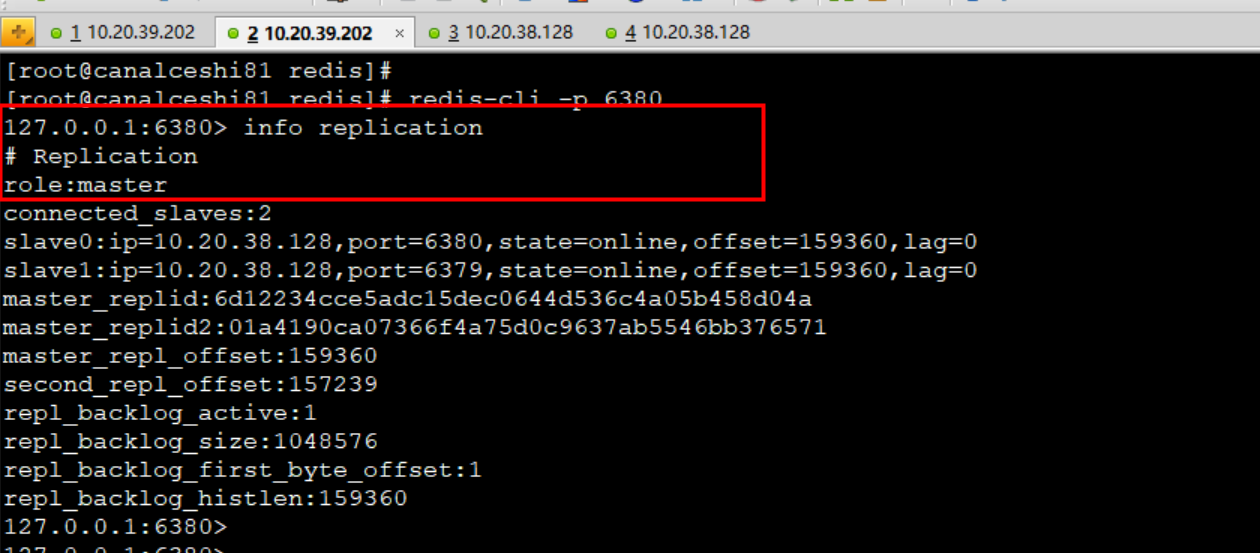
主节点存入数据，从节点读取数据；





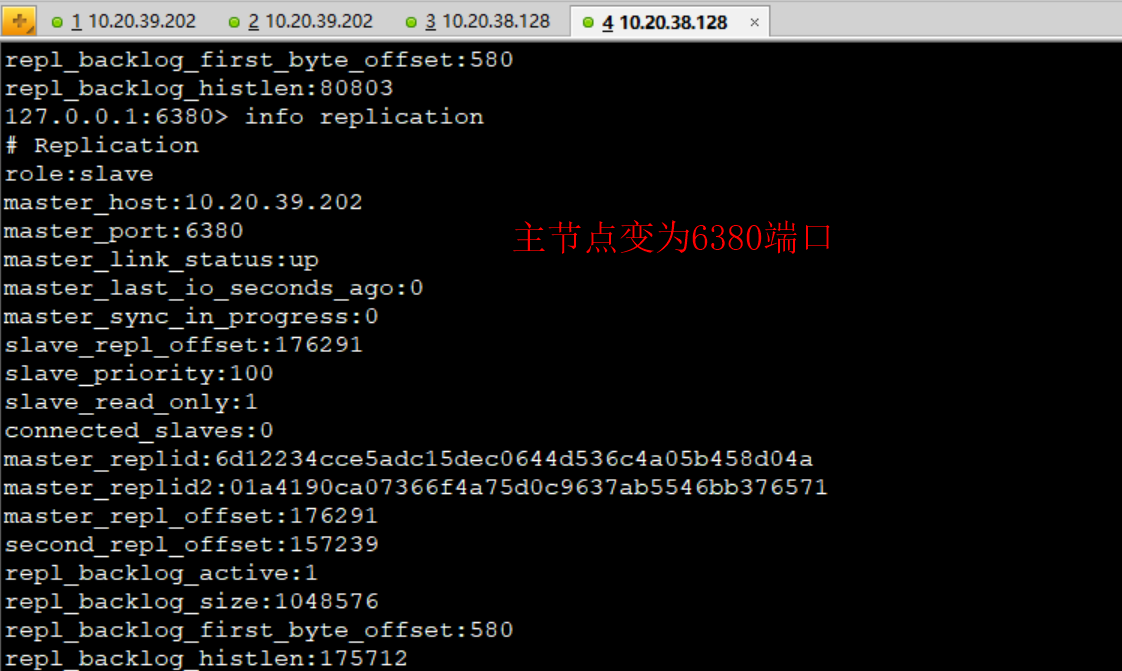
## 3、主机房主节点宕机

#### （1）主机房02机器变为主节点

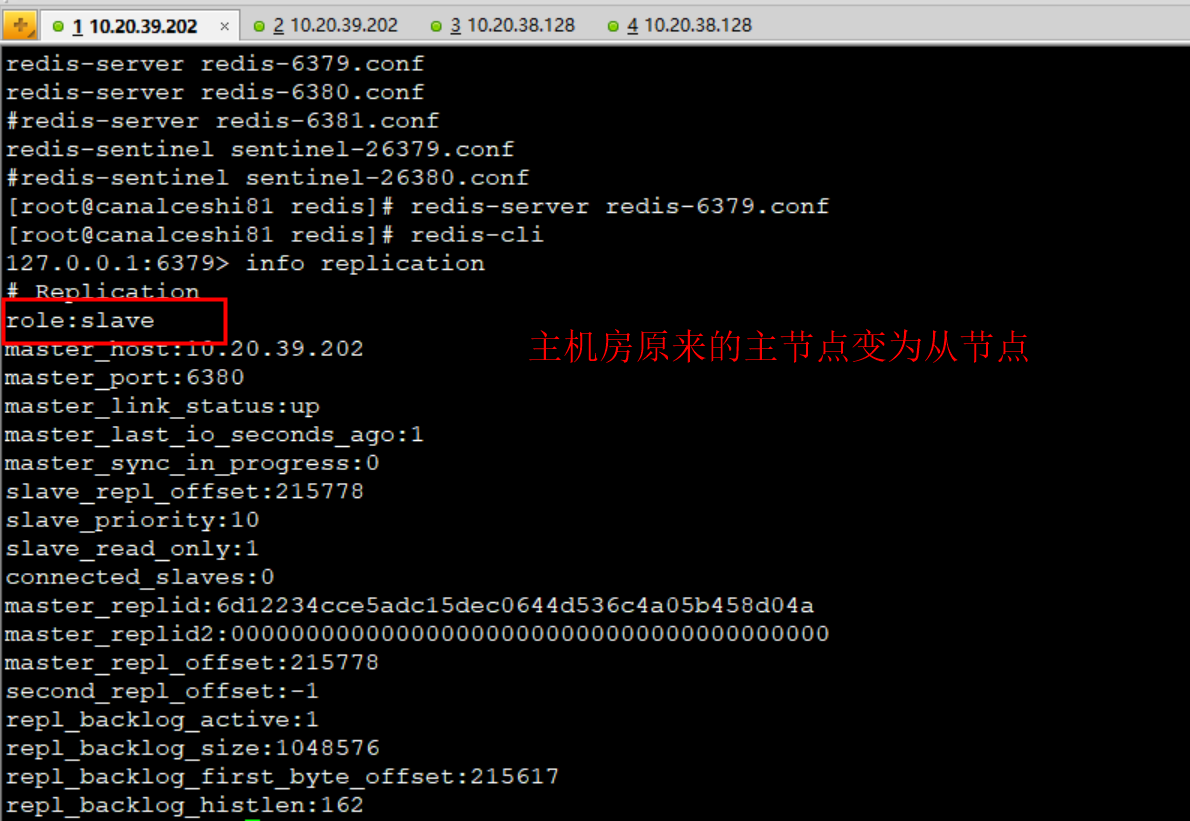


#### （1）备机房2台机器还是从节点



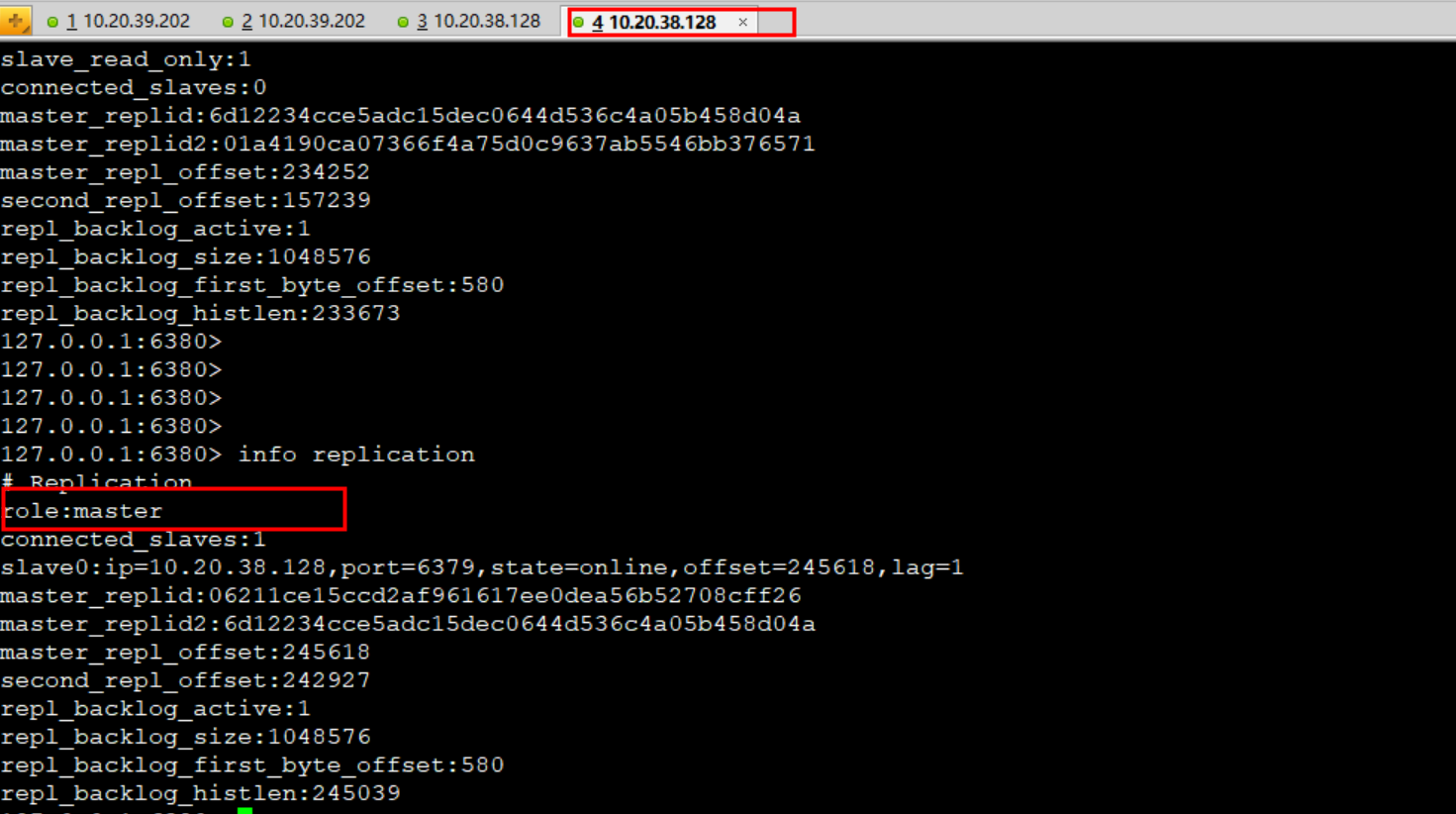


## 4、主机房主节点再次启动，主机房原来的主节点变为从节点

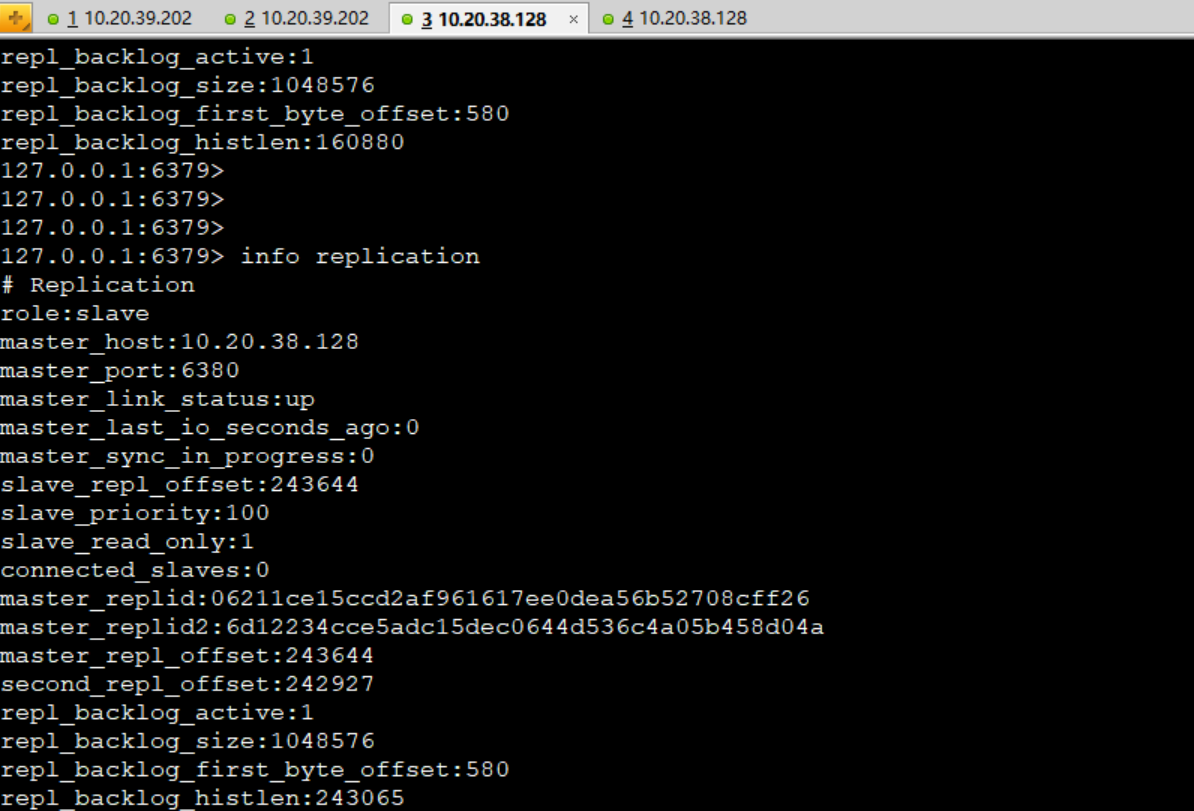


## 5、主机房全部宕机

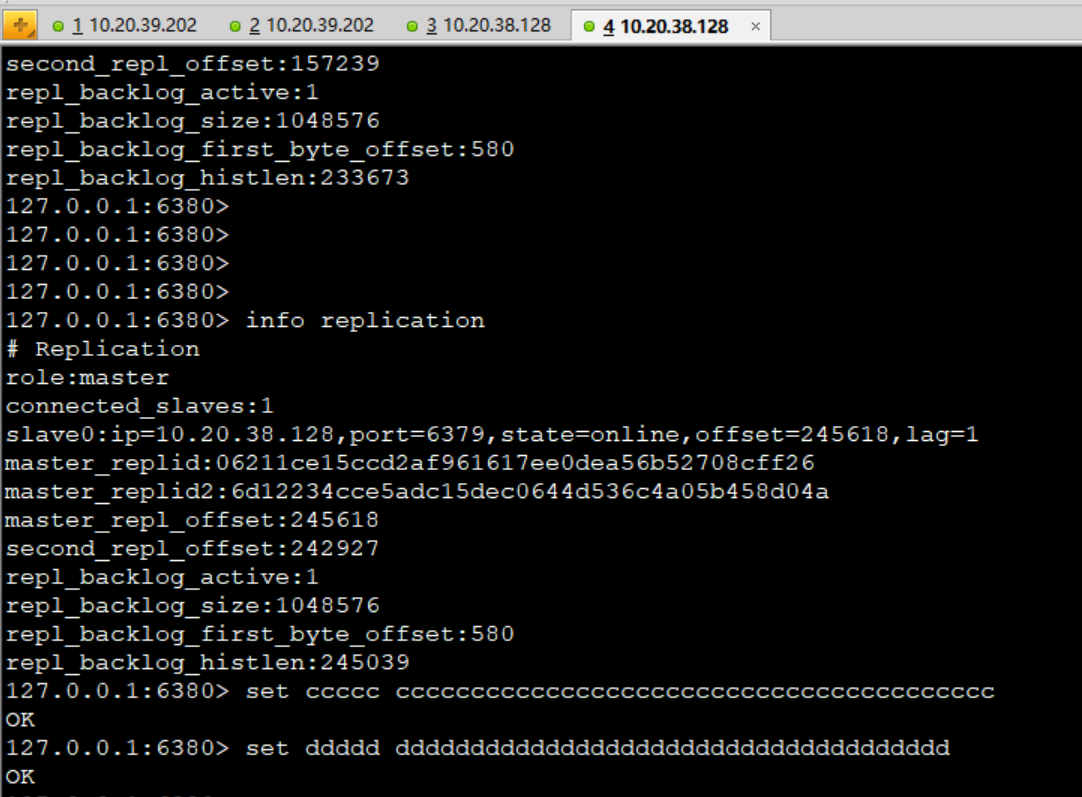
#### （1）备机房02机器瞬间变为主节点



#### （2）备机房01机器还是从节点



#### （3）备机房写入部分数据

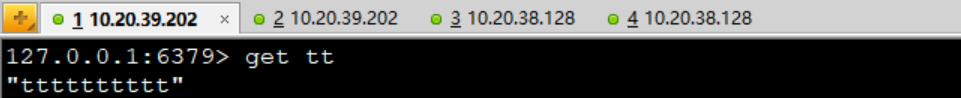


## 6、主机房恢复启动

#### （1）主节点还是在备机房

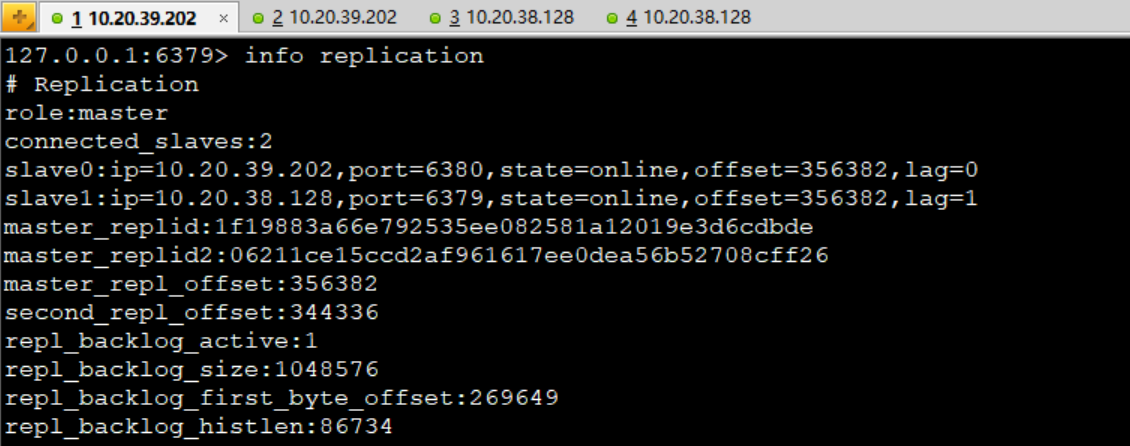


#### （2）数据可以同步会主机房

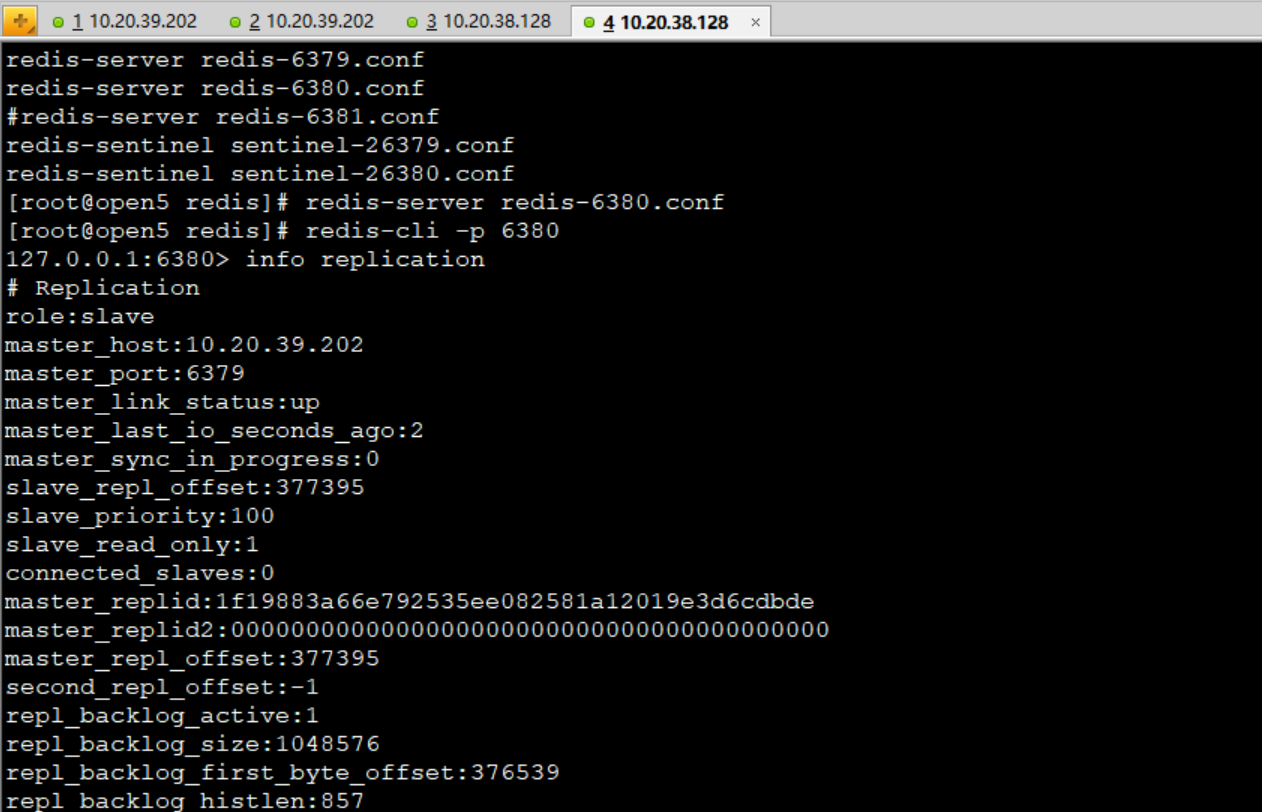


## 6、主节点切回主机房

#### （1）找到备机房的主节点，执行kill操作。30秒后，主节点切回主机房01机器



#### （2）再次启动备用机房挂掉的机器（原来备机房的主节点），此时这台机器变为从节点



# 四、高可用验证方案2：模拟机房断网（2个机房网络中断）



## 主机房与备用机房网络中断

当机房A与机房B网络中断后，机房B的哨兵会选举一个新的主节点出来，此时整个集群有2个主节点。机房内部的微服务继续可以对Redis读写。



## 主机房与备用机房网络恢复

当机房A与机房B网络恢复后，机房A的主节点变为从节点。机房B的Redis数据会同步到机房A。机房A中Redis数据会被机房B的Redis数据覆盖。

