# 一、如何搭建主从服务器

解决redis的两个问题：

高可用 主机挂了 从机能顶上

高并发

10万/秒

## 1.创建两个文件夹

一个是主文件夹

一个是从文件夹

在这两个文件夹内存放两个redis的配置文件以data数据卷、

## 2.使用docker来创建两个容器，并启动

一个是主服务器：6379

一个是从服务器：6380

## 3.在从服务器的配置文件中：加入密码的验证

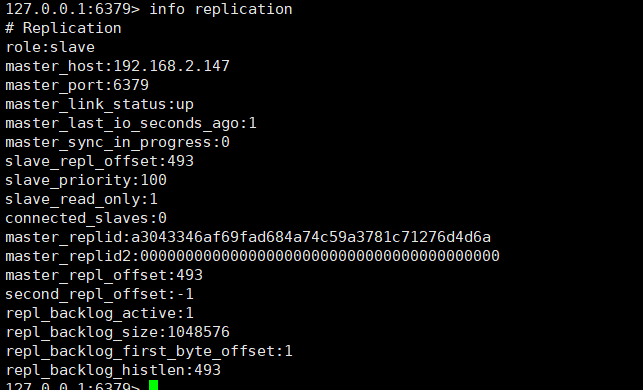
masterauth java1902

## 4.在从服务器中使用命令来连接主服务器

slaveof 主服务器的ip地址 端口号

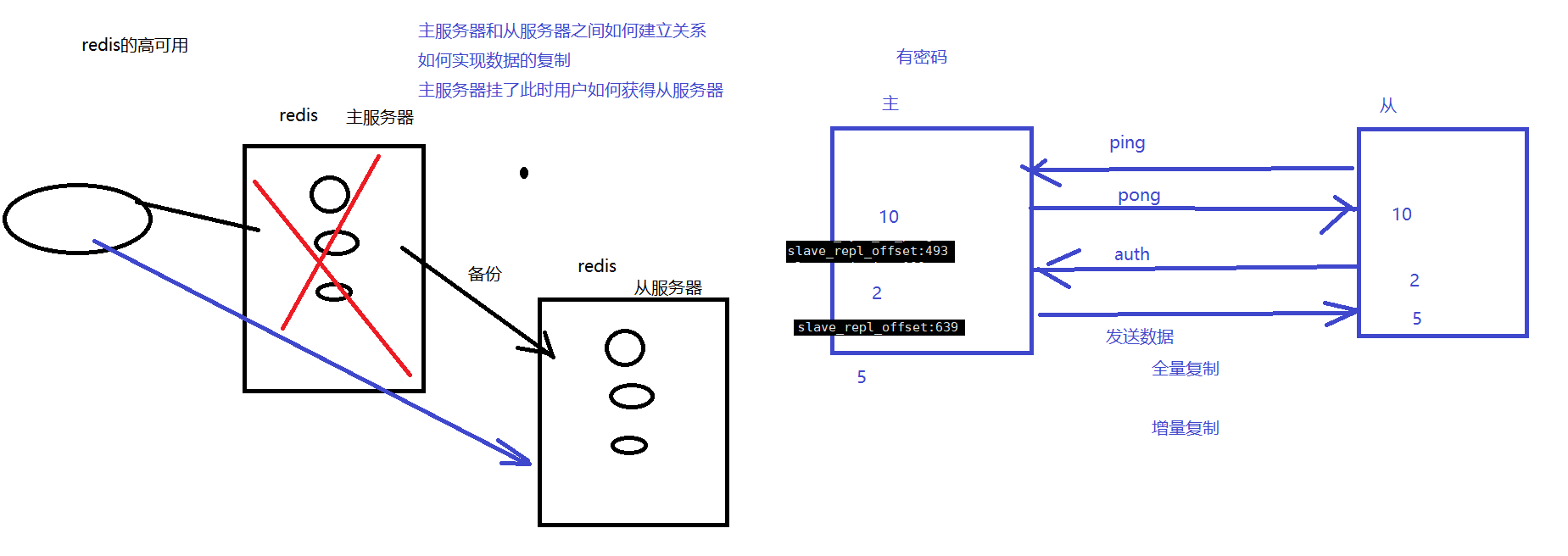
## 5.查看连接状态

info replication



一开始 做的是全量复制。

此时，主服务器根据从服务存储的数据的偏移量，来对新增的数据 增量复制给从服务器。



# 二、如何搭建哨兵集群

创建一台主服务器 创建一台从服务器

在从服务器中使用命令： slaveof 主服务器的ip 端口号

当主机挂了，从服务器顶上： 目前这个方案只能手动实现

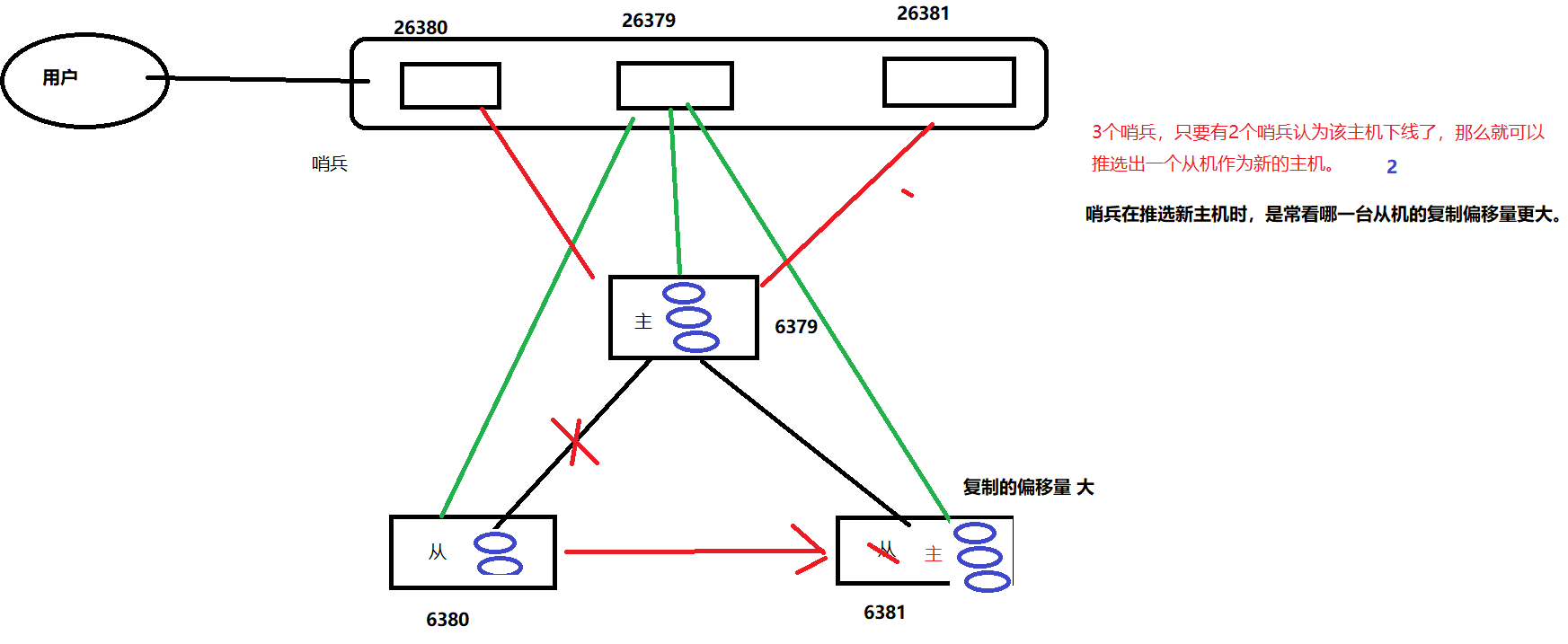
当主机恢复了， 主机要变成从机。 目前这个方案也只能手动实现。

其他的从机要连接新的主机 ： ....

用户需要获取一个新的ip地址： 手动来做

这些手动操作的方案有自动的工具能实现吗？

—— 哨兵



搭建1主2从3哨兵 redis集群的步骤

1）先搭建1主2从

让2个从服务器去连接主服务器

2）搭建3哨兵

|  |
| --- |
| docker run -v /usr/local/docker/redis-sentinel/sentinels/sentinel1.conf:/usr/local/etc/redis/sentinel.conf --name redis-sentinel1 -p 26379:26379 -d redis redis-sentinel /usr/local/etc/redis/sentinel.conf |

还需要注意，ip地址：

sentinel monitor mymaster 192.168.2.147 6379 2

要去配置主服务器。 密码， 超时时间等等...

# 三、Spring配置redis-sentinel

主要修改配置文件即可

|  |
| --- |
| *<!-- 哨兵配置 -->* <**bean id="sentinelConfig" class="org.springframework.data.redis.connection.RedisSentinelConfiguration"**>  *<!-- 配置master主服务的名称 -->* <**property name="master"**>  <**bean class="org.springframework.data.redis.connection.RedisNode"**>  <**property name="name" value="mymaster"**/>  </**bean**>  </**property**>  *<!-- 配置哨兵服务 -->* <**property name="sentinels"**>  <**set**>  <**bean class="org.springframework.data.redis.connection.RedisNode"**>  <**constructor-arg name="host" value="192.168.2.147"**/>  <**constructor-arg name="port" value="26379"**/>  </**bean**>  <**bean class="org.springframework.data.redis.connection.RedisNode"**>  <**constructor-arg name="host" value="192.168.2.147"**/>  <**constructor-arg name="port" value="26380"**/>  </**bean**>  <**bean class="org.springframework.data.redis.connection.RedisNode"**>  <**constructor-arg name="host" value="192.168.2.147"**/>  <**constructor-arg name="port" value="26381"**/>  </**bean**>  </**set**>  </**property**> </**bean**>     *<!-- 配置redis连接池对象 -->* <**bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig"**>  *<!-- 最大空闲数 -->* <**property name="maxIdle" value="50"** />  *<!-- 最大连接数 -->* <**property name="maxTotal" value="100"** />  *<!-- 最大等待时间 -->* <**property name="maxWaitMillis" value="20000"** /> </**bean**>  <**bean id="stringRedisSerializer" class="org.springframework.data.redis.serializer.StringRedisSerializer"**/>  *<!-- 配置redis连接工厂 -->* <**bean id="connectionFactory"  class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory"**>  *<!-- 连接池配置 -->* <**property name="poolConfig" ref="poolConfig"** />   *<!--哨兵配置-->* <**constructor-arg name="sentinelConfig" ref="sentinelConfig"**/>   *<!--&lt;!&ndash; 连接主机 &ndash;&gt;-->  <!--<property name="hostName" value="192.168.2.147" />-->  <!--&lt;!&ndash; 端口 &ndash;&gt;-->  <!--<property name="port" value="6379" />-->  <!-- 密码 -->* <**property name="password" value="java1902"** /> </**bean**>  *<!-- 配置redis模板对象 -->* <**bean class="org.springframework.data.redis.core.RedisTemplate"**>  *<!-- 配置连接工厂 -->* <**property name="connectionFactory" ref="connectionFactory"** />  *<!--配置序列化器-->* <**property name="keySerializer" ref="stringRedisSerializer"**/>  <**property name="valueSerializer" ref="stringRedisSerializer"**/> </**bean**> |

三、搭建Redis集群 解决Redis的高性能的问题

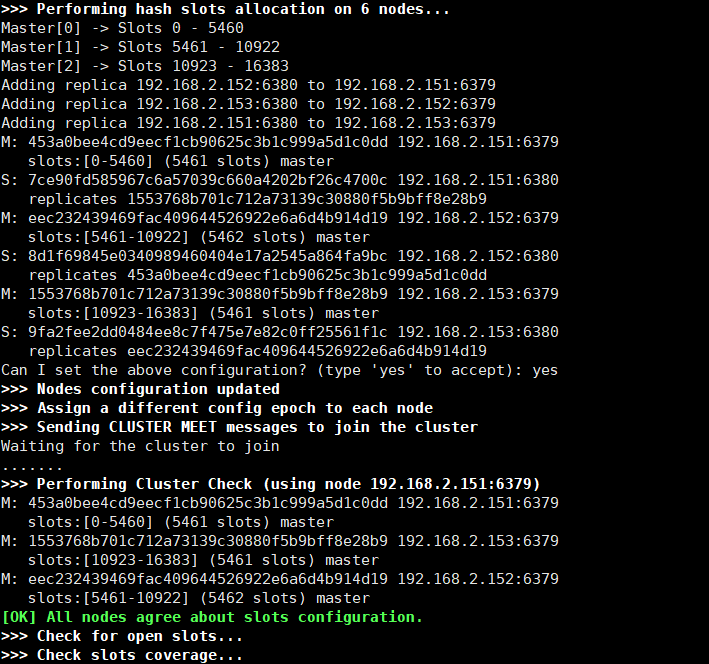
注意，在搭建时，一定不要先给密码。

还要把保护模式设为no

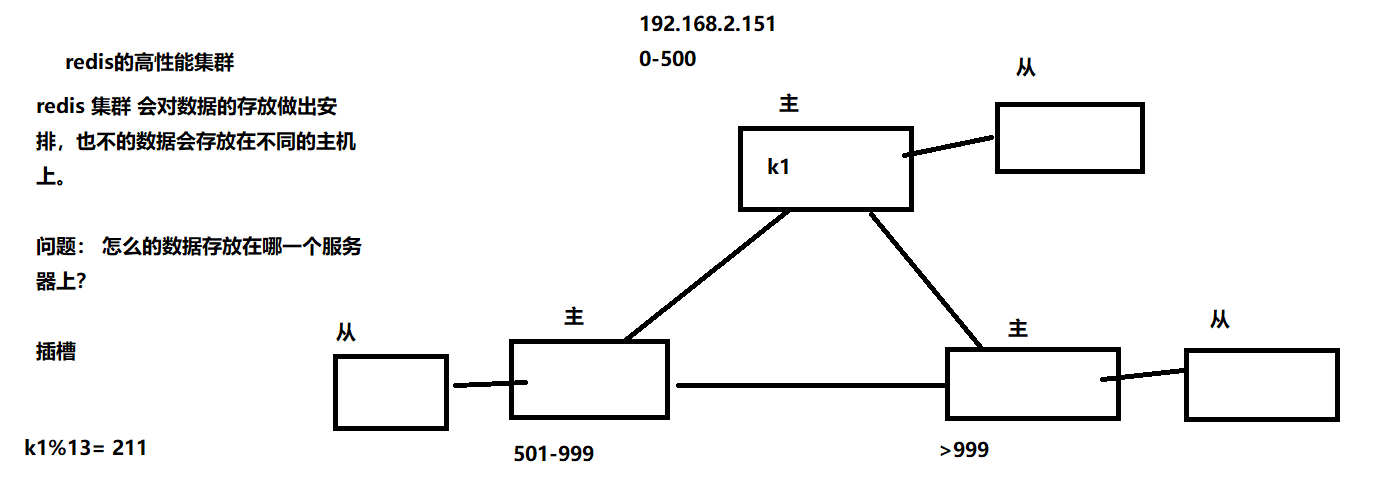
protected-mode no

注意：redis-cli提供的集群功能中，会自动帮我们建立主从关系。也就是说我们不需要自己建了。

搭建成功后的图



根据插槽的值来决定哪个数据放在哪个服务器上



搭建成功后的操作

注意，会从不同服务器上获得数据

