此文章已于 22:47:39 2017/11/28 发布到 HealerJean梦想博客

redis 主从复制

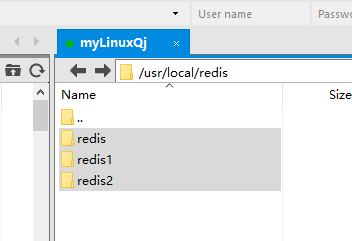
类别 服务器 ; Linux ; [选择一个类别或键入一个新类别]

## 主从复制

### 就是读写分离的意思，从主master数据库中添加，从slave数据库中读取内容，主的删除，从的就会自动删除(主数据库自动会向从数据库复制内容，也就是自动同步)

## 准备工作

### 需要准备3个redis数据库，分别配置redis.conf



# 1、主 数据库redis

## 1、解释：master:

### 1、port 端口为6000

### 2、requirepass密码设置为123456

|  |
| --- |
| port 6000  requirepass 123456 |

# 2、从数据库库redis分别为

## 解释：

#### 1、masterauth是从机对主机验证时，所需的密码。（即主机的requirepass）

#### 2、slaveof 表示主的地址和端口

## 1、slave1:

|  |
| --- |
| port 6001  slaveof 127.0.0.1 6000  masterauth 123456  requirepass 123456 |

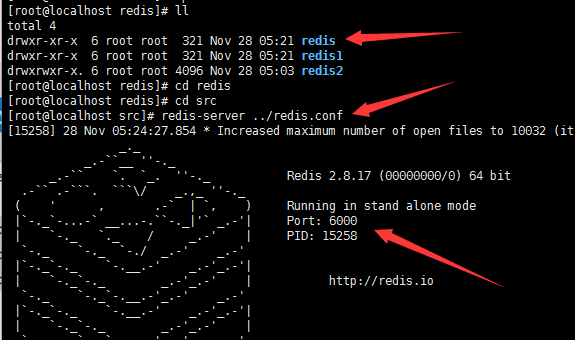
## 2、slave2:

|  |
| --- |
| port 6002  slaveof 127.0.0.1 6000  masterauth 123456  requirepass 123456 |

# 3、启动这三个redis

## 1、启动redis，根据配置文件redis.conf

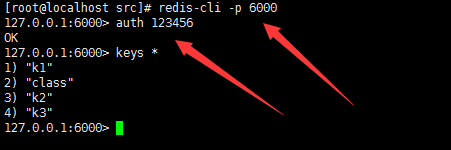
### redis-server ../redis.conf



## 2、连接这个6000的主服务器redis

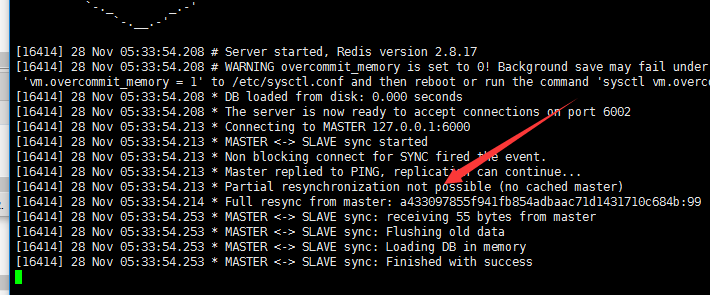
### redis-cli -p 6000 表示连接6000的端口的redis

### auth 123456 密码认证



## 3、其他的类似启动

### 1、从数据库redis，启动发现加载了master

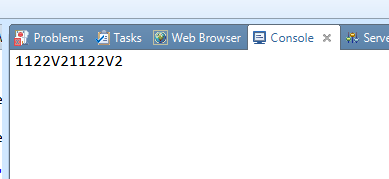


# 4、代码模拟测试

## 测试内容，给主数据库master 写入数据，但是从从数据库slave中读取

|  |
| --- |
| **public** **class** MaterSlaveMain {    @SuppressWarnings("resource")  **public** **static** **void** main(String[] args) {  //主从复制，配从不配主  Jedis jedis\_M = **new** Jedis("192.168.1.37",6000);  jedis\_M.auth("123456");  Jedis jedis\_S6001 = **new** Jedis("192.168.1.37",6001);  jedis\_S6001.auth("123456");  jedis\_S6001.slaveof("192.168.1.37",6000);  Jedis jedis\_S6002 = **new** Jedis("192.168.1.37",6002);  jedis\_S6002.auth("123456");  jedis\_S6002.slaveof("192.168.1.37",6000);    jedis\_M.set("class","1122V2");    String result1 = jedis\_S6001.get("class");  String result2 = jedis\_S6002.get("class");  System.*out*.println(result1+result2);  }  } |

## 2、测试结果，成功



# 3、添加哨兵，防止主master挂掉之后不能够实现主从复制，

## 注意事项： sentinel monitor 有系统默认的，需要先删掉，再添加，否则就重复了

## 1、主master redis中编辑sentinel.conf

### （在redis1中也编辑一个sentinel.conf，如2中所示，也就是说当master挂掉，之后，slave充当master）

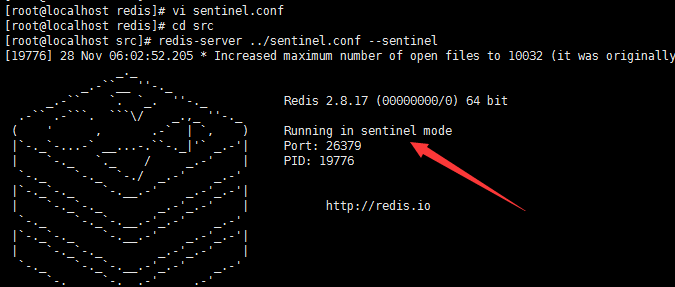
|  |
| --- |
| port 26379      sentinel monitor mymaster 127.0.0.1 6000 2    sentinel auth-pass mymaster 123456 |

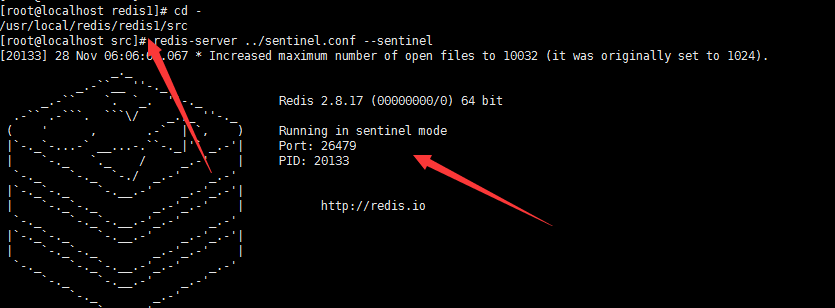
## 2、从slave redis1中，编辑sentinel.conf

|  |
| --- |
| port 26479    sentinel monitor mymaster 127.0.0.1 6000 2    sentinel auth-pass mymaster 123456 |

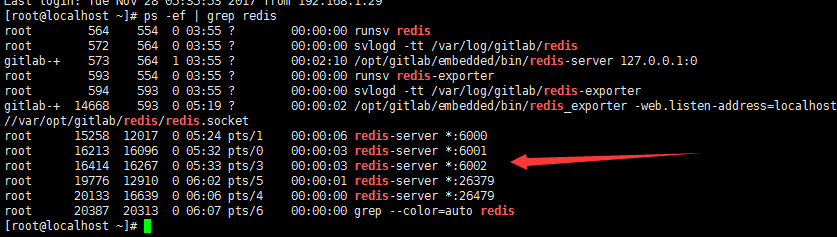
## 3、分别启动这两个sentinel.conf

### redis-server ../sentinel.conf --sentinel



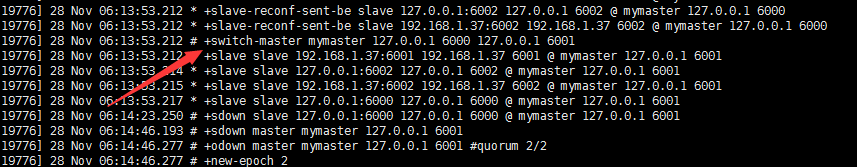


## 4、查看redis进程，哈哈，小伙伴们都在了



## 5、现在杀死6000 master的进程，观察master sentinel.conf中日志

### 1、会发现有个switch ，也就是说　master切换了，redis1变成了master



### 2、现在启动maser 6000，会发现，6001现在编程了master，牛逼了，居然还能修改配置文件，  因为sentinel在切换master的时候，把对应的sentinel.conf和redis.conf文件的配置修改。包括redis2，redis1，redis，全部都改，真牛



# 4、文件内容