**redis的主从复制(读写分离)/哨兵(主从切换)配置**

当数据量变得庞大的时候，读写分离还是很有必要的。同时避免一个redis服务宕机，导致应用宕机的情况，我们启用sentinel(哨兵)服务，实现主从切换的功能。redis提供了一个master,多个slave的服务。

         准备三个redis服务，依次命名文件夹子master,slave1,slave2.这里为在测试机上，不干扰原来的redis服务，我们master使用6000端口。

配置文件(redis.conf):

master配置修改端口：

port 6000

requirepass 123456

slave1修改配置：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | port 6001    slaveof 127.0.0.1 6000    masterauth 123456    requirepass 123456 |

slave2修改配置：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | port 6002    slaveof 127.0.0.1 6000    masterauth 123456    requirepass 123456 |

      requirepass：是认证密码，应该之后要作主从切换，所以建议所有的密码都一致。masterauth是从机对主机验证时，所需的密码(即主机的requirepass)。

启动主机：

|  |  |
| --- | --- |
| 1 | redis-server redis.conf |

启动从机：

redis-server redis1.conf

redis-server redis2.conf

产看启动进程，输入：

ps -ef |grep redis

root 6617 1 0 18:34 ? 00:00:01 redis-server \*:6000

root 6647 1 0 18:43 ? 00:00:00 redis-server \*:6001

root 6653 1 0 18:43 ? 00:00:00 redis-server \*:6002

root 6658 6570 0 18:43 pts/0 00:00:00 grep redis

可以看到主，从机的redis已经相应启动。

我们来验证下主从复制：

master:

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@localhost master]# redis-cli -p 6000  127.0.0.1:6000> auth 123456  OK  127.0.0.1:6000> set test chenqm  OK |

slave1:

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@localhost slave2]# redis-cli -p 6001  127.0.0.1:6001> auth 123456  OK  127.0.0.1:6001> get test  "chenqm" |

slave2:

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@localhost slave2]# redis-cli -p 6002  127.0.0.1:6002> auth 123456  OK  127.0.0.1:6002> get test  "chenqm" |

可以看到主机执行写命令，从机能同步主机的值，主从复制，读写分离就实现了。

但是万一主机挂了怎么办，这是个麻烦事情，所以redis提供了一个sentinel（哨兵），以此来实现主从切换的功能，类似与zookeeper。

Redis哨兵(sentinel)实现主从切换：

我们配置两个sentinel进程：

（1）

vi sentinel.conf

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | port 26379      sentinel monitor mymaster 127.0.0.1 6000 2    sentinel auth-pass mymaster 123456 |

（2）

|  |  |
| --- | --- |
| 1 | vi sentinel.conf |

|  |  |
| --- | --- |
| 1  2  3  4  5 | port 26479    sentinel monitor mymaster 127.0.0.1 6000 2    sentinel auth-pass mymaster 123456 |

启动sentinel服务（到对应的目录执行相应的命令）：

|  |  |
| --- | --- |
| 1 | redis-server sentinel.conf --sentinel |

 查看日志：

[7014] 11 Jan 19:42:30.918 # +monitor master mymaster 127.0.0.1 6000 quorum 2

[7014] 11 Jan 19:42:30.923 \* +slave slave 127.0.0.1:6002 127.0.0.1 6002 @ mymaster 127.0.0.1 6000

[7014] 11 Jan 19:42:30.925 \* +slave slave 127.0.0.1:6001 127.0.0.1 6002 @ mymaster 127.0.0.1 6000

从对应的日志观察到，一个master服务，两个slave服务。

我们现在来kill master进程

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | [root@localhost slave1]# ps -ef|grep redis  root      6960     1  0 19:29 ?        00:00:02 redis-server \*:6000  root      6968     1  0 19:30 ?        00:00:01 redis-server \*:6001  root      6975     1  0 19:30 ?        00:00:01 redis-server \*:6002  root      7014  6570  0 19:42 pts/0    00:00:01 redis-server \*:26479  root      7017  6789  0 19:42 pts/5    00:00:01 redis-server \*:26379  root      7021  6729  0 19:46 pts/3    00:00:00 grep redis  [root@localhost slave1]# kill -9 6960 |

我们观察日志：

|  |  |
| --- | --- |
| 1  2 | [7014] 11 Jan 19:43:41.463 # +sdown master mymaster 127.0.0.1 6000  [7014] 11 Jan 19:46:42.379 # +switch-master mymaster 127.0.0.1 6000 127.0.0.1 6001 |

master切换了，当6000端口的这个服务重启的时候，他会变成6001端口服务的slave。

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。

期间我们还需要关注的一个问题：sentinel服务本身也不是万能的，也会宕机，所以我们还得部署sentinel集群，象我这样多启动几个sentinel。

注意这个配置：

|  |  |
| --- | --- |
| 1 | sentinel monitor mymaster 127.0.0.1 6000 2   //这个后面的数字2,是指当有两个及以上的sentinel服务检测到master宕机，才会去执行主从切换的功能。 |