

二

## 38 实战：Redis 哨兵模式（下）

上一篇我们介绍了 Redis Sentinel 的搭建和运行原理，本文我们重点来看下 Sentinel 的命令操作和代码实战。

### Sentinel 命令操作

要使用 Sentinel 实现要连接到 Sentinel 服务器，和连接 Redis 服务相同，我们可以使用 redis-cli 来连接 Sentinel，如下命令所示：

```
[@iZ2ze0nc5n41zomzyqtksmZ:~]$ redis-cli -h 127.0.0.1 -p 26379 -a pwd654321
127.0.0.1:26379> ping
PONG
```

其中：

- -h 后面输入的是 Sentinel 的 IP；
- -p 后面输入的是 Sentinel 的端口，默认是 26379；
- -a 后面输入的是密码。

Sentinel 的端口号可以在 sentinel.conf 里面配置，通过 port 选项设置。

**注意：Sentinel 可以监视多台主节点，而不是只能监视一台服务器。**想要监视多台主节点只需要在配置文件中设置多个 `sentinel monitor master-name ip port quorum` 即可，我们通过 master-name 来区分不同的主节点。

### 查询所有被监控的主服务器信息

```
127.0.0.1:26379> sentinel masters
1) 1) "name"
   2) "mymaster"
   3) "ip"
   4) "127.0.0.1"
   5) "port"
```

```
6) "6377"
7) "runid"
8) "eb3552c6fc8974f91466c4ada90fe23ef30fd89c"
9) "flags"
10) "master"
11) "link-pending-commands"
12) "0"
13) "link-refcount"
14) "1"
15) "last-ping-sent"
16) "0"
17) "last-ok-ping-reply"
18) "400"
19) "last-ping-reply"
20) "400"
21) "down-after-milliseconds"
22) "30000"
23) "info-refresh"
24) "5731"
25) "role-reported"
26) "master"
27) "role-reported-time"
28) "75963321"
29) "config-epoch"
30) "7"
31) "num-slaves"
32) "2"
33) "num-other-sentinels"
34) "1"
35) "quorum"
36) "2"
37) "failover-timeout"
38) "180000"
39) "parallel-syncs"
40) "1"
```

相关语法:

```
sentinel masters
```

因为我们配置的 Sentinel 只监视了一台主服务器, 所以只有一台服务器的信息。

## 查询某个主节点的信息

```
127.0.0.1:26379> sentinel master mymaster
1) "name"
2) "mymaster"
3) "ip"
4) "127.0.0.1"
```

```
5) "port"
6) "6377"
7) "runid"
8) "eb3552c6fc8974f91466c4ada90fe23ef30fd89c"
9) "flags"
10) "master"
11) "link-pending-commands"
12) "0"
13) "link-refcount"
14) "1"
15) "last-ping-sent"
16) "0"
17) "last-ok-ping-reply"
18) "250"
19) "last-ping-reply"
20) "250"
21) "down-after-milliseconds"
22) "30000"
23) "info-refresh"
24) "8191"
25) "role-reported"
26) "master"
27) "role-reported-time"
28) "76096303"
29) "config-epoch"
30) "7"
31) "num-slaves"
32) "2"
33) "num-other-sentinels"
34) "1"
35) "quorum"
36) "2"
37) "failover-timeout"
38) "180000"
39) "parallel-syncs"
40) "1"
```

相关语法:

```
sentinel master master-name
```

## 查看某个主节点的 IP 和端口

```
127.0.0.1:26379> sentinel get-master-addr-by-name mymaster
1) "127.0.0.1"
2) "6377"
```

## 相关语法:

```
sentinel get-master-addr-by-name master-name
```

## 查询从节点的信息

```
127.0.0.1:26379> sentinel slaves mymaster #获取方式一
```

- 1) 1) "name"  
2) "127.0.0.1:6379"  
3) "ip"  
4) "127.0.0.1"  
5) "port"  
6) "6379"  
7) "runid"  
8) "14734d6065d745d89f115ca4735e7eeeeaa1a59b"  
9) "flags"  
10) "slave"  
11) "link-pending-commands"  
12) "0"  
13) "link-refcount"  
14) "1"  
15) "last-ping-sent"  
16) "0"  
17) "last-ok-ping-reply"  
18) "389"  
19) "last-ping-reply"  
20) "389"  
21) "down-after-milliseconds"  
22) "30000"  
23) "info-refresh"  
24) "390"  
25) "role-reported"  
26) "slave"  
27) "role-reported-time"  
28) "982798"  
29) "master-link-down-time"  
30) "1582192784000"  
31) "master-link-status"  
32) "err"  
33) "master-host"  
34) "127.0.0.1"  
35) "master-port"  
36) "6377"  
37) "slave-priority"  
38) "100"  
39) "slave-repl-offset"  
40) "1"
- 2) 1) "name"  
2) "127.0.0.1:6378"  
3) "ip"  
4) "127.0.0.1"

```
5) "port"
6) "6378"
7) "runid"
8) "f9d69479ace6c9eb4a6dfffa58ebc1ddf3de456e0"
9) "flags"
10) "slave"
11) "link-pending-commands"
12) "0"
13) "link-refcount"
14) "1"
15) "last-ping-sent"
16) "0"
17) "last-ok-ping-reply"
18) "390"
19) "last-ping-reply"
20) "390"
21) "down-after-milliseconds"
22) "30000"
23) "info-refresh"
24) "4004"
25) "role-reported"
26) "slave"
27) "role-reported-time"
28) "76212633"
29) "master-link-down-time"
30) "0"
31) "master-link-status"
32) "ok"
33) "master-host"
34) "127.0.0.1"
35) "master-port"
36) "6377"
37) "slave-priority"
38) "100"
39) "slave-repl-offset"
40) "10811245"
127.0.0.1:26379> sentinel replicas mymaster #获取方式二
1) 1) "name"
2) "127.0.0.1:6379"
3) "ip"
4) "127.0.0.1"
5) "port"
6) "6379"
7) "runid"
8) "14734d6065d745d89f115ca4735e7eeeeaa1a59b"
9) "flags"
10) "slave"
11) "link-pending-commands"
12) "0"
13) "link-refcount"
14) "1"
15) "last-ping-sent"
16) "0"
17) "last-ok-ping-reply"
18) "100"
19) "last-ping-reply"
```

```
20) "100"
21) "down-after-milliseconds"
22) "30000"
23) "info-refresh"
24) "100"
25) "role-reported"
26) "slave"
27) "role-reported-time"
28) "1071687"
29) "master-link-down-time"
30) "1582192873000"
31) "master-link-status"
32) "err"
33) "master-host"
34) "127.0.0.1"
35) "master-port"
36) "6377"
37) "slave-priority"
38) "100"
39) "slave-repl-offset"
40) "1"
2) 1) "name"
    2) "127.0.0.1:6378"
    3) "ip"
    4) "127.0.0.1"
    5) "port"
    6) "6378"
    7) "runid"
    8) "f9d69479ace6c9eb4a6dffa58ebc1ddf3de456e0"
    9) "flags"
    10) "slave"
    11) "link-pending-commands"
    12) "0"
    13) "link-refcount"
    14) "1"
    15) "last-ping-sent"
    16) "0"
    17) "last-ok-ping-reply"
    18) "100"
    19) "last-ping-reply"
    20) "100"
    21) "down-after-milliseconds"
    22) "30000"
    23) "info-refresh"
    24) "2496"
    25) "role-reported"
    26) "slave"
    27) "role-reported-time"
    28) "76301522"
    29) "master-link-down-time"
    30) "0"
    31) "master-link-status"
    32) "ok"
    33) "master-host"
    34) "127.0.0.1"
    35) "master-port"
```

```
36) "6377"
37) "slave-priority"
38) "100"
39) "slave-repl-offset"
40) "10823208"
```

相关语法:

```
sentinel replicas mymaster
```

或

```
sentinel slaves master-name
```

## 查询 Sentinel 集群中的其他 Sentinel 信息

```
127.0.0.1:26379> sentinel sentinels mymaster
1) 1) "name"
   2) "6455f2f74614a71ce0a63398b2e48d6cd1cf0d06"
   3) "ip"
   4) "127.0.0.1"
   5) "port"
   6) "26377"
   7) "runid"
   8) "6455f2f74614a71ce0a63398b2e48d6cd1cf0d06"
   9) "flags"
  10) "sentinel"
  11) "link-pending-commands"
  12) "0"
  13) "link-refcount"
  14) "1"
  15) "last-ping-sent"
  16) "0"
  17) "last-ok-ping-reply"
  18) "571"
  19) "last-ping-reply"
  20) "571"
  21) "down-after-milliseconds"
  22) "30000"
  23) "last-hello-message"
  24) "1043"
  25) "voted-leader"
  26) "?"
  27) "voted-leader-epoch"
  28) "0"
```

相关语法：

```
sentinel sentinels master-name
```

## 检查可用 Sentinel 的数量

```
127.0.0.1:26379> sentinel ckquorum mymaster  
OK 2 usable Sentinels. Quorum and failover authorization can be reached
```

有两个可用的 Sentinel，可用完成仲裁和故障转移授权。

相关语法：

```
sentinel ckquorum master-name
```

## 强制故障转移

```
127.0.0.1:26379> sentinel failover mymaster  
OK
```

相关语法：

```
sentinel failover master-name
```

## 在线修改配置信息

在 Redis 2.8.4 之前如果需要修改 Sentinel 的配置文件，例如添加或删除一个监视主节点，需要先停止 Sentinel 服务，再找到配置文件修改之后，重新启动 Sentinel 才行，这样就给我们带来了很大的不便，尤其是生产环境的 Sentinel，正常情况下如果是非致命问题我们是不能手动停止服务的，幸运的是 Redis 2.8.4 之后，我们可以不停机在线修改配置文件了，修改命令有以下几个。

## 增加监视主节点



使用 `sentinel monitor mymaster IP Port Quorum` 命令来添加监视主节点，如下命令所示：

```
127.0.0.1:26379> sentinel monitor mymaster 127.0.0.1 6379 2
OK
```

OK 表示添加监视主节点成功。

## 移除主节点的监视

使用 `sentinel remove master-name` 命令来实现移除主节点的监视，如下命令所示：

```
127.0.0.1:26379> sentinel remove mymaster
OK
```

OK 表示操作成功。

## 修改 quorum 参数

使用 `sentinel set master-name quorum n` 来修改 quorum 参数，如下命令所示：

```
127.0.0.1:26379> sentinel set mymaster quorum 1
OK
```

quorum 参数用来表示确认主节点下线的 Sentinel 数量，如果 quorum 设置为 1 表示只要有一台 Sentinel 确认主节点下线后，这个主节点就客观（真正地）下线了。

小贴士：以上所有对配置文件的修改，都会自动被刷新到物理配置文件 `sentinel.conf` 中。

## 代码实战

本文我们通过 Java 代码来实现，通过 Sentinel 连接信息获取相关 Redis 客户端，再进行相关 Redis 操作，这样 Sentinel 就会帮我们做容灾恢复，我们就不用担心操作某一个 Redis 服务器端，因为服务器挂了之后就会导致程序不可用了，具体实现代码如下：

```
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisSentinelPool;
import utils.Config;
```

```
import java.util.HashSet;
import java.util.Set;

public class SentinelExample {
    // master name
    private static String _MASTER_NAME = "mymaster";

    public static void main(String[] args) {
        // Sentinel 配置信息
        Set<String> set = new HashSet<>();
        // 连接信息 ip:port
        set.add("127.0.0.1:26379");
        // 创建 Sentinel 连接池
        JedisSentinelPool jedisSentinel = new JedisSentinelPool(_MASTER_NAME,
            set, Config.REDIS_AUTH);
        // 获取 Redis 客户端
        Jedis jedis = jedisSentinel.getResource();
        // 设置元素
        String setRes = jedis.set("key", "Hello, redis.");
        System.out.println(setRes);
        // 获取元素
        System.out.println(jedis.get("key"));
    }
}
```

以上程序执行结果如下:

```
OK
Hello, redis.
```

## 小结

本文我们讲了 Sentinel 相关的命令操作, 主要是用于查询相关主从节点和其他 Sentinel 信息的, 还可以执行强制故障转移等, 我们还讲了 2.8.4 提供的在线修改 Sentinel 参数的三个方法, 方便我们更好的使用 Sentinel, 最后用代码实现了通过 Sentinel 获取主节点并进行 Redis 服务器操作的实例, 这样就讲完整个 Sentinel 的介绍和应用。

[上一页](#)

[下一页](#)