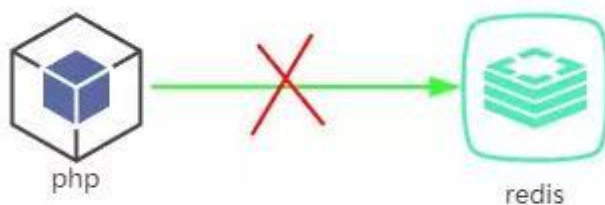


## 第 23 课

# Redis Sentinel 实现 redis 高可用

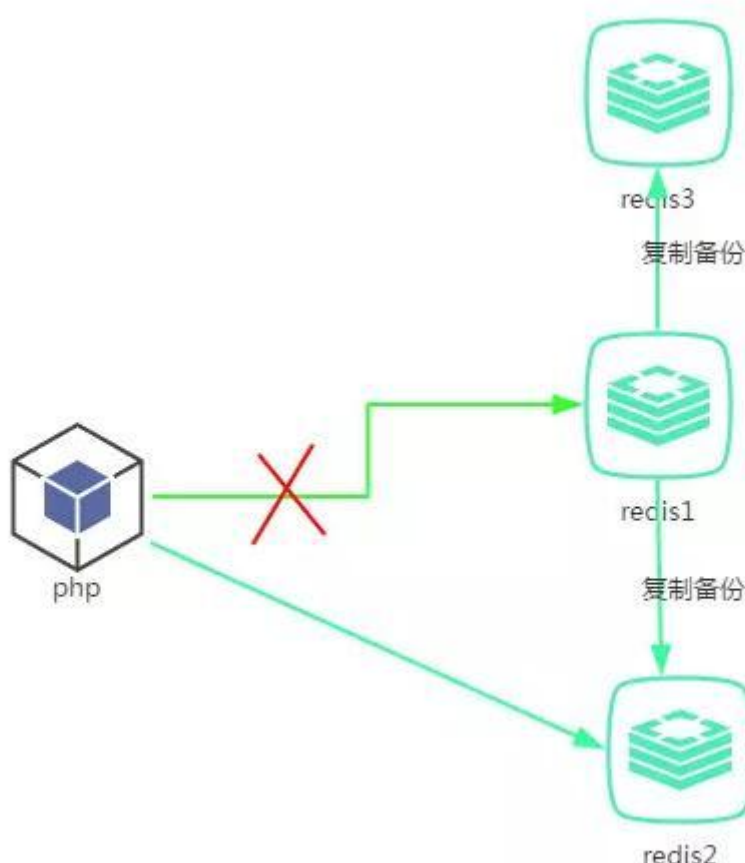
### 一、单实例模式

当系统中只有一台 redis 运行时，一旦该 redis 挂了，会导致整个系统无法运行。



### 二、主从模式

由于单台 redis 出现单点故障，就会导致整个系统不可用，所以想到的办法自然就是备份。当一台 redis 出现问题了，另一台 redis 可以继续提供服务。



### 三、自动故障转移机制

虽然上面 redis 做了备份，看上去很完美。但由于 redis 目前只支持主从复制备份（不支持主主复制），当主 redis 挂了，从 redis 只能提供读服务，无法提供写服务。所以，还得想办法，当主 redis 挂了，让从 redis 升级成为主 redis。

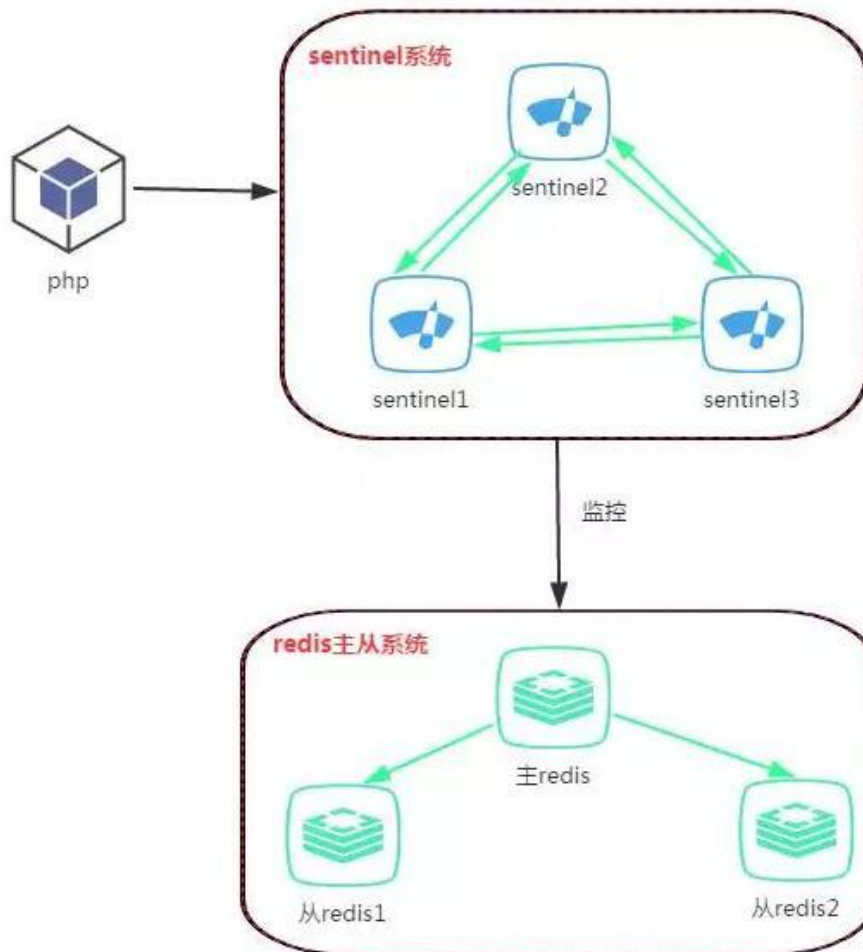
这就需要自动故障转移，redis sentinel 带有这个功能，当一个主 redis 不能提供服务时，redis sentinel 可以将一个从 redis 升级为主 redis，并对其他从 redis 进行配置，让它们使用新的主 redis 进行复制备份。

Redis-Sentinel 是 Redis 官方推荐的高可用性(HA)解决方案，当用 Redis 做 Master-slave 的高可用方案时，假如 master 宕机了，Redis 本身(包括它的很多客户端)都没有实现自动进行主备切换，而 Redis-sentinel 本身也是一个独立运行的进程，它能监控多个 master-slave 集群，发现 master 宕机后能进行自动切换。它的主要功能有以下几点

实时地监控 redis 是否按照预期良好地运行;

如果发现某个 redis 节点运行出现状况，能够通知另外一个进程(例如它的客户端);

能够进行自动切换。当一个 master 节点不可用时，能够选举出 master 的多个 slave(如果有超过一个 slave 的话)中的一个来作为新的 master,其它的 slave 节点会将它所追随的 master 的地址改为被提升为 master 的 slave 的新地址。



## 四、高可用 redis 部署

### 1.环境描述

这里使用三台服务器，每台服务器上开启一个 redis-server 和 redis-sentinel 服务，redis-server 端口为 8000，redis-sentinel 的端口为 6800，修改默认端口是安全的第一步。

redis-server 说明

172.16.213.234:8000 主  
172.16.213.233:8000 从  
172.16.213.232:8000 从

redis-sentinel 说明

172.16.213.234:6800  
172.16.213.233:6800  
172.16.213.232:6800

## 2.搭建 redis 系统

首先下载安装 redis

```
wget http://download.redis.io/releases/redis-3.2.9.tar.gz
tar zxvf redis-3.2.9.tar.gz
cd redis-3.2.9
make
```

```
cd src
```

```
#复制 redis 相关命令到/usr/sbin 目录下，这样就可以直接执行这些命令，不用写全路径
cp redis-cli redis-server redis-sentinel /usr/sbin/
```

在 redis 目录下有 redis.conf 和 sentinel.conf 配置文件示例，拷贝 redis.conf、sentinel.conf 到 /opt/redis（当然也可以在/etc/目录新建配置文件），然后修改配置文件。

修改主 redis-server 配置文件内容如下：

```
port 8000
daemonize yes
bind 0.0.0.0pidfile /var/run/redis-8000.pid
logfile /var/log/redis/redis-8000.log
```

修改从 redis-server 配置文件内容如下：

```
port 8000
daemonize yes
bind 0.0.0.0pidfile
/var/run/redis-8000.pid
logfile /var/log/redis/redis-8000.log
slaveof 172.16.213.234 8000 #从 redis 比主 redis 多这一行
```

启动 redis-server。

```
redis-server /opt/redis/redis.conf
```

三个 redis 服务启动完毕后，进入命令行，执行 info replication 查看当前主从配置。

## 3.搭建 redis-sentinel 系统

redis-sentinel 程序上面已经安装过了，这里只需要修改配置文件就可以了。修改 /opt/redis/sentinel.conf，如果没有创建即可。

修改 sentinel.conf 配置文件内容如下:

```
daemonize yes
port 6800
logfile /var/log/redis/sentinel.log
pidfile /var/run/sentinel.pid
sentinel monitor master8000 172.16.213.234 8000 2
sentinel down-after-milliseconds master8000 5000
sentinel failover-timeout master8000 15000
```

启动 redis-sentinel:

```
redis-sentinel /opt/redis/sentinel.conf
```

三个 redis-sentinel 服务启动完毕后, 连接任意 sentinel 服务可以获知当前主 redis 服务信息。

## 五、测试

1. 把住 redis 停掉

```
redis-cli -h 172.16.213.234 -p 8000 shutdown
```

2. 查看 redis-sentinel 的监控状态

此时会发现, 172.16.213.233 这台 redis-server 提升为主库。

至此, redis 的高可用方案已经搭建完成。

## 六、客户端问题

客户端程序 (如 PHP 程序) 连接 redis 时需要 ip 和 port, 但 redis-server 进行故障转移时, 主 redis 是变化的, 所以 ip 地址也是变化的。客户端程序如何感知当前主 redis 的 ip 地址和端口呢?

### VIP 方案

VIP 方案是, redis 系统对外始终是同一 ip 地址, 当 redis 进行故障转移时, 需要做的是将 VIP 从之前的 redis 服务器漂移到现在新的主 redis 服务器上。

比如: 当前 redis 系统中主 redis 的 ip 地址是 172.16.213.234, 那么 VIP (172.16.213.238) 指向 172.16.213.234, 客户端程序用 VIP (172.16.213.238) 地址连接 redis, 实际上连接的就是当前主 redis, 这样就避免了向 sentinel 发送请求。

当主 redis 宕机, 进行故障转移时, 172.16.213.233 这台服务器上的 redis 提升为主, 这时 VIP (172.16.213.238) 指向 172.16.213.233, 这样客户端程序不需要修改任何代码,

连接的是 172.16.213.233 这台主 redis。

## 七、漂移 VIP

那么现在的问题是，如何在进行 redis 故障转移时，将 VIP 漂移到新的主 redis 服务器上。

这里可以使用 redis sentinel 的一个参数 client-reconfig-script，这个参数配置执行脚本，sentinel 在做 failover 的时候会执行这个脚本，并且传递 6 个参数 <master-name>、<role>、<state>、<from-ip>、<from-port>、<to-ip>，其中 <to-ip> 是新主 redis 的 IP 地址，可以在这个脚本里做 VIP 漂移操作。

```
sentinel client-reconfig-script master8000 /opt/redis/notify_master6800.sh
```

修改三个服务器的 redis-sentinel 配置文件 /opt/redis/sentinel.conf，增加上面一行。然后在 /opt/redis 目录下创建 notify\_master6800.sh 脚本文件，这个脚本做 VIP 漂移操作，内容如下：

#notify\_master6800.sh 脚本内容

```
#!/bin/bash
MASTER_IP=$6 #第六个参数是新主 redis 的 ip 地址
LOCAL_IP='172.16.213.234' #其他两个服务器上为 172.16.213.233, 172.16.213.232
VIP='172.16.213.238'
NETMASK='24'
INTERFACE='eth1'
if [[ "${MASTER_IP}" == "${LOCAL_IP}" ]];then
    /sbin/ip addr add ${VIP}/${NETMASK} dev ${INTERFACE} #将 VIP 绑定到该服务器上
    /sbin/arping -q -c 3 -A ${VIP} -I ${INTERFACE}
    exit 0
else
    /sbin/ip addr del ${VIP}/${NETMASK} dev ${INTERFACE} #将 VIP 从该服务器上删除
    exit 0
fi
exit 1 #如果返回 1，sentinel 会一直执行这个脚本
```

现在当前主 redis 是 172.16.213.234，**第一次**，需要手动绑定 VIP 到该服务器上。

```
/sbin/ip addr add 172.16.213.238/24 dev eth1
/sbin/arping -q -c 3 -A 172.16.213.238 -I eth1
```

然后，去另一个服务器上通过 VIP 地址连接 redis-server 和 redis-sentinel。