redis+keepalived

环境说明

node1 centos6.7 192.168.96.1 关闭防火墙和selinux master

node2 centos6.7 192.168.96.2 关闭防火墙和selinux slave

node3 centos6.7 192.168.96.3 关闭防火墙和selinux slave

软件准备

redis-3.2.0.tar.gz

keepalived-1.2.19.tar.gz

软件安装

node1 2 3

```
tar zxf redis-3.2.8.tar.gz
cd redis-3.2.8
yum install gcc -y #源码编译安装需要C语言编译器，所以需要安装gcc
make
make install
./utils/install_server.sh    #自动配置 包括端口配置文件等等
```

服务配置

经过上述操作，在三台主机上安装了redis的服务。但是由于redis本身的安全性配置，默认情况下是只允许通过本机连接的。

通过telnet 127.0.0.1 6379可以发现，redis的服务是可以使用的。但是telnet 192.168.96.1 6379会发现拒绝访问。

这是因为在redis中有一个bind参数，通过此参数限制了可以连接的IP。

在配置文件中，通过添加bind ip来添加允许连接的地址。

需要注意的是，在3.2版本以前，不通过bind置顶任何地址代表允许任意地址链接。

但是在3.2版本开始，redis新增了保护机制，不指定地址的话，会默认拒绝所有链接，如果想要通过不指定地址的方式进行配置，需要将protected-mode参数关闭，即由yes改为no，以此关闭保护模式。

上述操作是必不可少的。

node2 3

```
echo 'slaveof 192.168.96.1 6379' >> /etc/redis/6379.conf
/etc/init.d/redis_6379 restart
```

通过此方式，将两个slave端和master端绑定。

此时，可以通过下述方式来查询集群状态。

```
redis-cli -h IP info replication  #IP为想要查询状态的redis地址，如192.168.96.1
```

如果是master，会显示其余加入集群的slave的信息。

如果是slave，会显示master的连接时间或下线时间。

通过此方法搭建的集群为主从集群，只有master对数据库的权限是读写，slave的权限为只读。

通过redis-sentinel服务实现主从切换

node1 2 3

```
vim /etc/redis/sentinel.conf
    port 26379
    daemonize yes
    protected-mode no
    logfile "/home/redis/logs/sentinel.log"
    sentinel announce-ip 192.168.0.201
    sentinel monitor mymaster 192.168.0.201 6379 2
    sentinel down-after-milliseconds mymaster 15000
    sentinel failover-timeout mymaster 900000
    sentinel parallel-syncs mymaster 1
redis-sentinel  /etc/redis/sentinel.conf
```

因为在安装redis的同时，已经安装了redis-sentinel。我们只需要自己手动的去创建或修改配置文件即可。

配置文件含义：

port 26379 ： 服务端口

daemonize yes ： 是否开启守护进程，yes为开启。

protected-mode no ： 同样的保护开关。此条与6379.conf的配置文件并不相同。

6379.conf中，保护模式是用来保护他人对数据库的链接，如果开启，则非指定的地址无法连接数据库。

此文件中的保护开关则不然。通过关闭此开关，才能将slave节点提升为master节点。

如果开启此保护，则集群会一直停留在等待master重连的状态，无法实现主从切换。

logfile "/home/redis/logs/sentinel.log" ： 定义日志文件

sentinel announce-ip 192.168.96.1 ： 定义master节点

sentinel monitor mymaster 192.168.96.1 6379 2 ： 将master节点命名为mymaster。

在集群工作是，会不同的检测master，即192.168.96.1 6379。

当有两个slave投票认为master异常，将会产生一个新的master。

sentinel down-after-milliseconds mymaster 15000 ： 异常时间判定 15000ms

即当超过15000ms无法连接master，则改节点会认为master异常。

sentinel failover-timeout mymaster 900000 ： 选举超时时间 900000ms

即当判定master失效后，如果超过设定时间为选出新的master，则选举失败。

试验中未出现超时，无法确定超时后会产生的后果。

sentinel parallel-syncs mymaster 1 ： 选举后同步服务器个数

即在新的选举结束后，最多有多少服务器对新的master进行同步。

个数越小则整个集群同步完成时间越慢，但是对master来说，压力也会越小。

redis-sentinel /etc/redis/sentinel.conf命令则是代表通过/etc/redis/sentinel.conf文件启动redis-sentinel服务。

redis-sentinel服务还有另外一种启动方法： redis-server /etc/redis/sentinel.conf --sentinel

通过上述配置，当master节点出现故障时，集群会重新选举一个slaver成为新的master节点，以保证服务。

但是就如同上文所说，此种方式组成的集群只有健康检查，无法将故障节点重启并重新加入集群。

还有另外需要注意的是，此种方法选举出新的master之后，原master手动加入集群后，并不会重新成为master，而是会作为slave存在。

通过keepalived实现vip自动转移

node 1 2 3

```
tar zxf keepalived-1.2.19.tar.gz
cd keepalived-1.2.19
./configure --prefix=/usr/local/keepalived
make
make install
cp /usr/local/keepalived/sbin/keepalived /usr/sbin/
cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
mkdir /etc/keepalived
```

node 1

```
vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
   notification_email {
     root@localhost
   }
   notification_email_from keepalived@localhost
   smtp_server 127.0.0.1
   smtp_connect_timeout 10
   router_id keepalivedha_1
}
vrrp_script chk_http_port {
    script "redis-cli info | grep role:master >/dev/null 2>&1"
    interval 1
    timeout 2
    fall 2
    rise 1
}
vrrp_sync_group VG_1 {
```

```
    group {
        VI_1
    }
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    #use_vmac keepalived
    #vmac_xmit_base
    mcast_src_ip 192.168.96.1
    smtp_alert
    virtual_router_id 20
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass password
    }
    virtual_ipaddress {
        192.168.96.4
    }
    track_script {
        chk_http_port
    }
}
service keepalived start
```

node 2

```
vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
   notification_email {
     root@localhost
   }
   notification_email_from keepalived@localhost
   smtp_server 127.0.0.1
   smtp_connect_timeout 10
   router_id keepalivedha_2
}
vrrp_script chk_http_port {
    script "redis-cli info | grep role:master >/dev/null 2>&1"
    interval 1
    timeout 2
    fall 2
```

```
    rise 1
}
vrrp_sync_group VG_1 {
    group {
        VI_1
    }
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    #use_vmac keepalived
    #vmac_xmit_base
    mcast_src_ip 192.168.96.2
    smtp_alert
    virtual_router_id 20
    priority 99
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass password
    }
    virtual_ipaddress {
        192.168.96.4
    }
    track_script {
        chk_http_port
    }
    nopreempt
}
service keepalived start
```

node 3

```
vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
   notification_email {
     root@localhost
   }
   notification_email_from keepalived@localhost
   smtp_server 127.0.0.1
   smtp_connect_timeout 10
   router_id keepalivedha_3
}
vrrp_script chk_http_port {
```

```
    script "redis-cli info | grep role:master >/dev/null 2>&1"
    interval 1
    timeout 2
    fall 2
    rise 1
}
vrrp_sync_group VG_1 {
    group {
        VI_1
    }
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    #use_vmac keepalived
    #vmac_xmit_base
    mcast_src_ip 192.168.96.3
    smtp_alert
    virtual_router_id 20
    priority 98
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass password
    }
    virtual_ipaddress {
        192.168.96.4
    }
    track_script {
        chk_http_port
    }
    nopreempt
}
service keepalived start
```