

~~~~~命令补充~~~~~

## 1. rpm

这里以tree

```
[root@localhost ~]# rpm -ql tree      # 查询
[root@localhost ~]# rpm -e tree      # 卸载
[root@localhost ~]# rpm -qa tree      # 列出所有安装过的包
帮助手册:https://www.cnblogs.com/ftl1012/p/rpm.html
```

## 2.haproxy的卸载

```
[root@localhost ~]# rpm -e haproxy
```

```
[root@localhost ~]# rpm -ql haproxy
package haproxy is not installed
```

```
[root@localhost ~]# find / -name haproxy
/var/lib/haproxy
```

```
[root@localhost lib]# rm -rf /var/lib/haproxy
```

## 3.haproxy的开启关闭

```
[root@localhost haproxy.cfg]# ps -ef |grep haproxy
haproxy  13045      1  0 15:52 ?          00:00:00 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg
root     13052     7529  0 15:53 pts/1    00:00:00 grep --color=auto haproxy
```

```
[root@localhost haproxy.cfg]# kill -9 13045
```

## 一.关于mysql连接的负载均衡

## 1.1配置文件内容:

```
global
log      127.0.0.1 local2
chroot   /var/lib/haproxy
pidfile  /var/run/haproxy.pid
maxconn  4000
user     haproxy
group    haproxy
daemon
stats socket /var/lib/haproxy/stats

defaults
mode                tcp
log                 global
option              tcplog
option              dontlognull
option http-server-close
option              redispatch
retries             3
timeout http-request 10s
timeout queue       1m
timeout connect     10s
timeout client      1m
timeout server      1m
timeout http-keep-alive 10s
timeout check       10s
maxconn             3000

frontend mysql
bind                0.0.0.0:3066
mode                tcp
log                 global
```

```
default_backend      mysql_server
```

```
backend      mysql_server
```

```
balance      roundrobin
```

```
server mysql1 192.168.100.149:3306 check inter 5s rise 2 fall 3
```

```
server mysql2 192.168.100.148:3306 check inter 5s rise 2 fall 3
```

```
listen stats
```

```
mode http
```

```
bind 0.0.0.0:1080
```

```
stats enable
```

```
stats hide-version
```

```
stats uri /haproxyadmin?stats
```

```
stats realm Haproxy\ Statistics
```

```
stats auth admin:admin
```

```
stats admin if TRUE
```

### 1.2 远程连接测试

<http://192.168.100.142:1080/haproxyadmin?stats>(根据你的实际情况来定)

### 测试

```
mysql -h 127.0.0.1 -P 3306 -uwill -p123456 --default_auth=mysql_native_password -e "show variables like 'server_id' ";
```

```
+-----+-----+
```

```
| Variable_name | Value |
```

```
+-----+-----+
```

```
| server_id     | 2     |
```

```
+-----+-----+
```

```
mysql -h 127.0.0.1 -P 3306 -uwill -p123456 --default_auth=mysql_native_password -e "show variables like 'server_id' ";
```

```
+-----+-----+
```

```
| Variable_name | Value |
```

```
+-----+-----+
```

```
| server_id     | 1     |
```

```
+-----+-----+
```

### 1.3 laravel的连接mycat负载均衡

修改.env文件:

```
DB_CONNECTION=mysql
```

```
DB_HOST=192.168.100.142
```

```
DB_PORT=8806
```

```
DB_DATABASE=TESTDB
```

```
DB_USERNAME=will
```

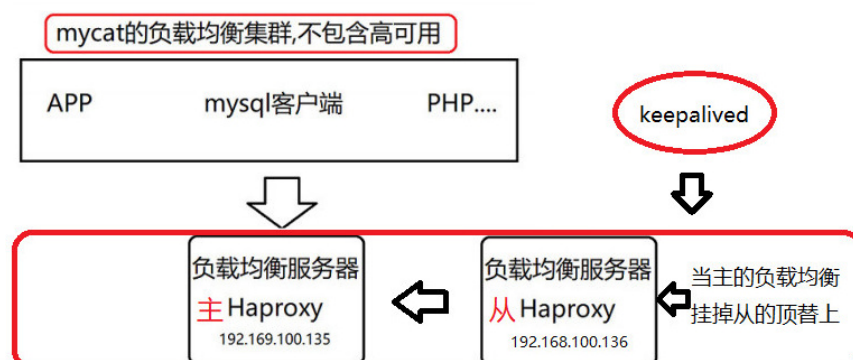
```
DB_PASSWORD=123456
```

出现报错:修改以下/config/database.php文件

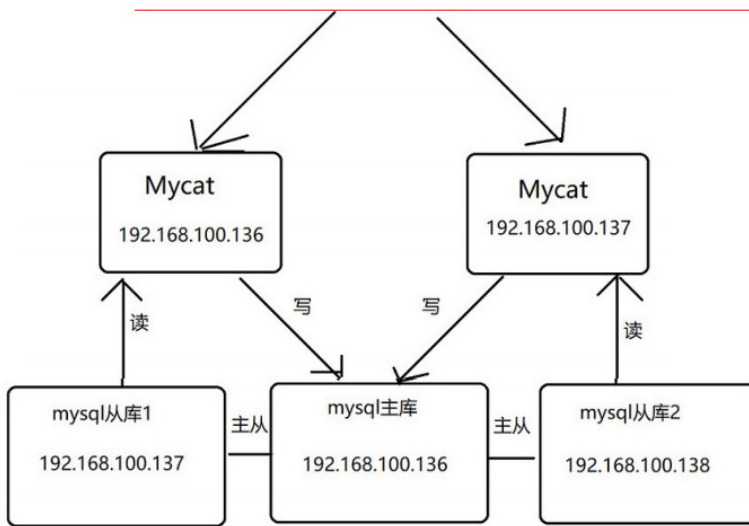
'strict' => true, //关闭严格模式,修改为false

## 二. Keepalived介绍

Keepalived的作用是检测服务器的状态,如果有一台服务器宕机,或工作出现故障,Keepalived将检测到,并将有故障的服务器从系统中剔除,同时使用其它服务器代替该服务器的工作,当服务器工作正常后Keepalived自动将服务器加入到服务器群中,这些工作全部自动完成,不需要人工干涉,需要人工做的只是修复故障的服务器。







### 三. Keepalived工具安装

#### 2.1 安装 keepalived 需要用到 openssl

方式一 yum安装:

```
[root@localhost ~]# yum install -y curl gcc openssl-devel libnl3-devel net-snmp-devel
[root@localhost ~]# yum install epel-release -y
[root@localhost ~]# yum install keepalived -y #通过yum直接安装
```

初始化及启动

```
[root@localhost ~]# systemctl start keepalived //启动 keepalived
[root@localhost ~]# systemctl stop keepalived //关闭 keepalived
[root@localhost ~]# systemctl enable keepalived //加入开机启动 keepalived
[root@localhost ~]# systemctl restart keepalived //重新启动 keepalived
[root@localhost ~]# systemctl status keepalived //查看 keepalived 状态
```

快速卸载

```
[root@localhost ~]# rpm -e keepalived
```

配置文件路径为/etc/keepalived/keepalived.conf

方式二 源码包安装:

```
[root@localhost ~]# yum install gcc gcc-c++ openssl openssl-devel
[root@localhost ~]# wget -q https://www.keepalived.org/software/keepalived-1.2.18.tar.gz
```

解压Keepalived并安装

```
[root@localhost ~]# tar -zxvf keepalived-1.2.18.tar.gz
[root@localhost ~]# cd keepalived-1.2.18
[root@localhost ~]# ./configure --prefix=/usr/local/keepalived
[root@localhost ~]# make && make install
```

将 keepalived 安装成 Linux 系统服务(因为没有使用 keepalived 的默认路径安装(我们使用的是/usr/local),安装完成之后,需要做一些工作 复制默认配置文件到默认路径)

```
[root@localhost ~]# mkdir /etc/keepalived
[root@localhost ~]# cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/
```

复制 keepalived 服务脚本到默认的地址

```
[root@localhost ~]# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
[root@localhost ~]# cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
[root@localhost ~]# ln -s /usr/local/keepalived/sbin/keepalived /usr/sbin/
```

设置 keepalived 服务开机启动

```
[root@localhost ~]# chkconfig keepalived on
```

此时的开关机命令

```
[root@localhost ~]# service strat keepalived
[root@localhost ~]# service stop keepalived
[root@localhost ~]# service restrat keepalived
```

## 2.2 卸载

#源码所在目录（根据实际情况进入对应目录）

```
[root@localhost keepalived-1.2.18]# ll
total 488
-rw-rw-r--. 1 1000 1000    41 Apr 25  2015 AUTHOR
drwxrwxr-x. 2 1000 1000   4096 May 14 05:11 bin
-rw-rw-r--. 1 1000 1000 158219 Jun 30  2015 ChangeLog
-rw-r--r--. 1 root root   25410 May 14 05:09 config.log
-rwxr-xr-x. 1 root root   25472 May 14 05:09 config.status
-rwxrwxr-x. 1 1000 1000 172872 May  9  2015 configure .....
执行命令：
[root@localhost keepalived-1.2.18]# make uninstall
查找相关文件依次删除
[root@localhost keepalived-1.2.18]# find / -name keepalived
```

## 四.Keepalived配置文件

```
global_defs {
    router_id LVS_MASTER
}

vrrp_sync_group VG1 {
    group {
        VI_1
    }
}

vrrp_script chk_haproxy {
    script "/etc/keepalived/haproxy_check.sh" ## 检测 haproxy 状态的脚本路径
    interval 2 ## 检测时间间隔
    weight 2 ## 如果条件成立，权重+2
}

vrrp_instance VI_1 {
    state MASTER
    interface ens33
    virtual_router_id 51
    priority 100
    advert_int 1
    nopreempt
    authentication {
        auth_type PASS
        auth_pass 1234
    }
    virtual_ipaddress {
        192.168.199.100
    }
}

vrrp_instance VI_1 {
    state MASTER
    interface ens33
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.199.101
    }
}

virtual_server 192.168.199.100 3306 {
```

```

# 定义虚拟服务器，地址与上面的virtual_ipaddress相同
delay_loop 3 # 健康检查时间间隔，3秒
lb_algo rr # 负载均衡调度算法: rr|wrr|lc|wlc|sh|dh|lbc
lb_kind DR # 负载均衡转发规则: NAT|DR|TUN
#persistence_timeout 5 # 会话保持时间5秒，动态服务建议开启
protocol TCP # 转发协议protocol，一般有tcp和udp两种
real_server 192.168.199.142 3306 {
    weight 1 # 权重越大负载分越大，0表示失效
    TCP_CHECK {
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
        connect_port 3306
    }
}
}
track_script {
    chk_haproxy
}

```

运行脚本:

```

[root@localhost keepalived-1.2.18]# vi /etc/keepalived/haproxy_check.sh
#!/bin/bash
START_HAPROXY="/usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg" #haproxy启动命令
LOG_FILE="/usr/local/keepalived/log/haproxy-check.log" # 日志文件
HAPS=`ps -C haproxy --no-header |wc -l` # 检测haproxy的状态，0代表未启动,1已经启动
date "+%Y-%m-%d %H:%M:%S" >> $LOG_FILE #在日志文件当中记录检测时间
echo "check haproxy status" >> $LOG_FILE # 记录haproxy的状态
if [ $HAPS -eq 0 ];then #执行haproxy判断
echo $START_HAPROXY >> $LOG_FILE #记录启动命令
/usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg #启动haproxy
sleep 3
if [ `ps -C haproxy --no-header |wc -l` -eq 0 ];then
echo "start haproxy failed, killall keepalived" >> $LOG_FILE
killall keepalived
service keepalived stop
fi
fi

为脚本增加执行权限: chmod +x haproxy_check.sh

```