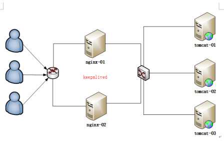
# keepalived

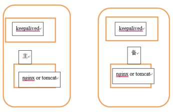
## 1. 简介

Keepalived的作用是检测web服务器的状态，如果有一台web服务器死机，或工作出现故障，Keepalived将检测到，并将有故障的web服务器从系统中剔除，当web服务器工作正常后Keepalived自动将web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web服务器。 Keepalived是一款可以实现高可靠的软件，通常部署在2台服务器上，分为一主一备。Keepalived可以对本机上的进程进行检测，一旦Master检测出某个进程出现问题，将自己切换成Backup状态，然后通知另外一个节点切换成Master状态。



## 2. 作用

主要用作RealServer的健康状态检查以及LoadBalance主机和BackUP主机之间failover的实现。



## 3. 安装

### 1. keepalived安装

下载keepalived官网:http://keepalived.org

1. 安装依赖 su - root yum -y install kernel-devel yum -y install openssl-  yum -y install popt-devel yum -y install lrzsz yum -y install openssh-clients

### 2. 将keepalived解压到/usr/local/src目录下

```
tar -zxvf keepalived-1.2.19.tar.gz -C /usr/local/src
```

### 3. 进入到/usr/local/src/keepalived-1.2.19目录

```
cd /usr/local/src/keepalived-1.2.19
```

### 4. 开始configure

```
./configure --prefix=/usr/local/keepalived
```

### 编译并安装

```
make && make install
```

## 4. 将keepalived添加到系统服务中

### 1. 拷贝执行文件

```
cp /usr/local/keepalived/sbin/keepalived /usr/sbin/
```

### 2. 将init.d文件拷贝到etc下,加入开机启动项

```
cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/keepalived
```

### 3. 将keepalived文件拷贝到etc下

```
cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
```

### 4. 创建keepalived文件夹

```
mkdir -p /etc/keepalived
```

### 5. 将keepalived配置文件拷贝到etc下

```
cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf
```

### 6. 添加可执行权限

```
chmod +x /etc/init.d/keepalived
```

### 7. 添加keepalived到开机启动

```
chkconfig --add keepalived
chkconfig keepalived on
```

8. 以上所有命令一次性执行：

```
cp /usr/local/keepalived/sbin/keepalived /usr/sbin/ cp /usr/local/keepalived/etc/rc.d/init.d/keepalived
/etc/init.d/keepalived cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/ mkdir -p
/etc/keepalived cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf
chmod +x /etc/init.d/keepalived chkconfig --add keepalived
chkconfig keepalived on
```

# 5. 配置keepalived虚拟IP

修改配置文件： /etc/keepalived/keepalived.conf

## MASTER节点

global_defs { notification_email {#指定keepalived在发生切换时需要发送email到的对象，一行一个

## acassen@firewall.loc

## failover@firewall.loc

## sysadmin@firewall.loc

} notification_email_from Alexandre.Cassen@firewall.loc#指定发件人

## smtp_server 192.168.200.1#指定smtp服务器地址

## smtp_connect_timeout 30 #指定smtp连接超时时间

router_id LVS_DEVEL#运行keepalived机器的一个标识 } vrrp_instance VI_1 { state MASTER #指定A节点为主节点 备用节点上设置为BACKUP即可 interface eth0 #绑定虚拟IP的网络接口 virtual_router_id 51 #VRRP组名，两个节点的设置必须一样，以指明各个节点属于同一VRRP组 priority 100 #主节点的优先级（1-254之间），备用节点必须比主节点优先级低 advert_int 1 #组播信息发送间隔，两个节点设置必须一样 authentication { #设置验证信息，两个节点必须一致 auth_type PASS auth_pass 1111 } virtual_ipaddress { #指定虚拟IP, 两个节点设置必须一样 192.168.33.60/24 #如果两个nginx的ip分别是192.168.33.61,,...62，则此处的虚拟ip跟它俩同一个网段即可 } virtual_server 192.168.56.70 8080 { delay_loop 6#健康检查时间间隔 lb_algo rr #调度算法rr|wrr|lc|wlc|lblc|sh|dh lb_kind DR #负载均衡转发规则NAT|DR|RUN

## nat_mask 255.255.255.0 #需要验证

persistence_timeout 1#会话保持时间 protocol TCP#使用的协议 real_server 192.168.56.201 8080 { weight 10 #默认为1,0为失效 SSL_GET { url { #检查url，可以指定多个 path / digest ff20ad2481f97b1754ef3e12ecd3a9cc #检查后的摘要信息 } url { path /mrtg/ digest 9b3a0c85a887a256d6939da88aabd8cd } connect_timeout 3#连接超时时间 nb_get_retry 3#重连次数 delay_before_retry 3#重连间隔时间 } }

### BACKUP节点

global_defs { } vrrp_instance VI_1 { state BACKUP interface eth0 virtual_router_id 51 priority 99 advert_int 1 authentication { auth_type PASS auth_pass 1111 } virtual_ipaddress { 192.168.33.60/24 } }

## 6. 分别启动两台机器上的keepalived

service keepalived start 测试：
杀掉master上的keepalived进程，你会发现，在slave机器上的eth0网卡多了一个ip地址 查看ip地址的命令：
ip addr

## 7. 配置keepalived心跳检查

原理： Keepalived并不跟nginx耦合，它俩完全不是一家人
但是keepalived提供一个机制：让用户自定义一个shell脚本去检测用户自己的程序，返回状态给keepalived就可以了

```
#MASTER??
global_defs {
}
vrrp_script chk_health {
    script "[[ `ps -ef | grep nginx | grep -v grep | wc -l` -ge 2 ]] && exit 0 || exi
t 1"
    interval 1    #??1?????????????????ngnix
    weight -2
}
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 1
    priority 100
    advert_int 2
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {
        chk_health
    }

    virtual_ipaddress {
        10.0.0.10/24
    }
    notify_master "/usr/local/keepalived/sbin/notify.sh master"
    notify_backup "/usr/local/keepalived/sbin/notify.sh backup"
    notify_fault "/usr/local/keepalived/sbin/notify.sh fault"
}
```

## 添加切换通知脚本

vi /usr/local/keepalived/sbin/notify.sh

# !/bin/bash

case "$1" in master) /usr/local/nginx/sbin/nginx exit 0 ;; backup) /usr/local/nginx/sbin/nginx -s stop /usr/local/nginx/sbin/nginx exit 0 ;; fault) /usr/local/nginx/sbin/nginx -s stop exit 0 ;; *) echo 'Usage: notify.sh {master|backup|fault}' exit 1 ;; esac

## 添加执行权限

chmod +x /usr/local/keepalived/sbin/notify.sh

```
global_defs {
}
vrrp_script chk_health {
    script "[[ `ps -ef | grep nginx | grep -v grep | wc -l` -ge 2 ]] && exit 0 || exi
t 1"
    interval 1
    weight -2
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 1
    priority 99
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    track_script {
        chk_health
    }

    virtual_ipaddress {
        10.0.0.10/24
    }
    notify_master "/usr/local/keepalived/sbin/notify.sh master"
    notify_backup "/usr/local/keepalived/sbin/notify.sh backup"
    notify_fault "/usr/local/keepalived/sbin/notify.sh fault"
}
```

## 在第二台机器上添加notify.sh脚本

## 分别在两台机器上启动keepalived

service keepalived start chkconfig keepalived on