

- 1
- 2 这个名字好难
- 3 因为我们视野的不开阔，导致缺乏解决问题的手段
- 4

keepalived nginx 双机热备图文讲解

原文：[http://blog.csdn.net/wanglei\\_storage/article/details/51175418](http://blog.csdn.net/wanglei_storage/article/details/51175418)

keepalived nginx 双机热备实战精讲

在试验之前先贴张图上来，大致说下具体的需求及环境（注：实验所用到的脚本在文章末尾，如果在过程中遇到可以先看脚本）

环境：

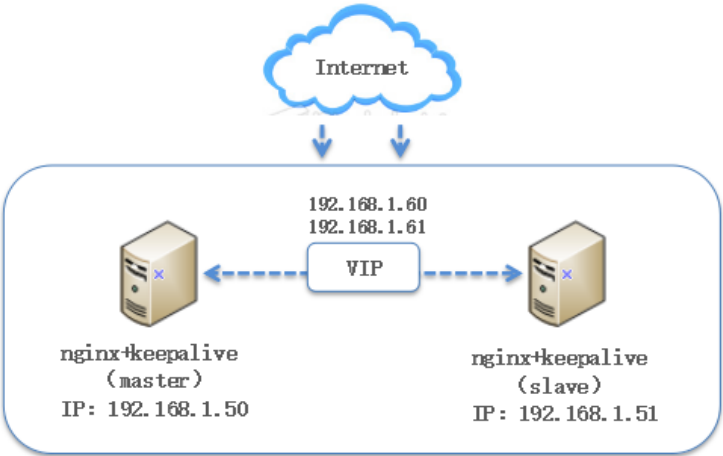
VIP（漂移地址）：192.168.1.60-192.168.1.61  
nginx-1：192.168.1.50  
nginx-2：192.168.1.51

关闭 iptables、关闭selinux 并且打通ssh配置，使得192.168.1.50和192.168.1.51可以免密码登录（主要同于同步 nginx 配置和 reload，公钥验证文件：authorized\_keys）

需求：

两只 nginx web 服务器分别提供相同的访问，当一只主机无法提供服务（keepalived、nginx 或者说主机宕机）后，另一只马上转换为MASTER状态，并且继承VIP，继续向用户提供服务。

- 1、如果 keepalived 状态转换为 MASTER，则发送邮件进行通知运维人员（由下面 keepalived\_check.sh 脚本实现）
- 2、需要nginx两台服务器提供的服务是完全一致的，并且当一台 nginx 的配置改变，则另一台立刻变为相同的配置，并且 reload。（由 rsync+inotify 实时同步脚本 nginx\_rsync.sh 脚本实现）
- 3、如果keepalived 挂掉，则检测并且重启keepalived；如果nginx 挂掉，则关闭keepalived，且让VIP转移到 keepalived-slave上面（由 nginx\_check.sh脚本实现）



下面开始实验：

一、nginx-1（master 主机配置）

- 1、时间校准  
ntpdate ntp.api.bz
- 2、安装依赖软件  
yum -y install openssl openssl-devel kernel-devel



昵称：[这个名字想了很久~](#)  
园龄：[2年6个月](#)  
粉丝：[167](#)  
关注：[8](#)  
[+加关注](#)

2019年2月						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

搜索

找我看

谷歌搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

我的标签

- [mysql\(60\)](#)
- [spring-boot\(44\)](#)
- [redis\(28\)](#)
- [nginx\(27\)](#)
- [maven\(25\)](#)
- [mybatis\(24\)](#)
- [openfire\(19\)](#)

- 3、使用inotify机制，还需要安装inotify-tools工具，以便提供inotifywait,inotifywatch辅助工具，用来监控，汇总改动情况
- yum -y install inotify-tools
- 4、准备 keepalived 和 nginx 包（keepalived 包可以去 keepalived.org 下载，nginx 包则去 tengine 官方下载）

linux(18)

eclipse(14)

spring(14)

更多

## 随笔档案

- 2019年2月 (6)
- 2019年1月 (12)
- 2018年12月 (14)
- 2018年11月 (11)
- 2018年10月 (8)
- 2018年9月 (14)
- 2018年8月 (58)
- 2018年7月 (15)
- 2018年6月 (3)
- 2018年5月 (4)
- 2018年4月 (12)
- 2018年3月 (49)
- 2018年2月 (10)
- 2018年1月 (7)
- 2017年12月 (9)
- 2017年11月 (14)
- 2017年10月 (36)
- 2017年9月 (11)
- 2017年8月 (46)
- 2017年7月 (16)
- 2017年6月 (63)
- 2017年5月 (31)
- 2017年4月 (24)
- 2017年3月 (15)
- 2017年2月 (26)
- 2017年1月 (81)
- 2016年12月 (72)
- 2016年11月 (70)
- 2016年10月 (69)
- 2016年9月 (49)
- 2016年8月 (56)
- 2016年7月 (1)

## 最新评论

1. Re:win10怎么彻底关闭自动更新  
还是会重新启动更新服务  
--Rest探路者
2. Re:(windows)一台电脑上安装两个Mysql服务  
字这么小，二维码倒不小  
--土了个豆
3. Re:SpringBoot集成JWT实现token验证  
如果在html的js 写一个异步请求，如何把token携带到请求里面？  
--QH.Thomas
4. Re:Linux下安装jdk8步骤详述  
非常详尽 一步一步按照教程来的,搞定了,非常感谢  
--MedusaSTears
5. Re:Linux下安装jdk8步骤详述  
扫了红包了还是自己用掉吧哈哈看你那边多少看你运气咯  
--632657030

## 阅读排行榜

1. Linux下安装jdk8步骤详述(222005)
2. linux查看内存占用情况(114649)
3. Linux下查看MySQL的安装路径(65599)
4. 更改MySQL数据库的编码为utf8mb4(59828)
5. mysql解决datetime与timestamp精确到毫秒的问题(47615)

## 评论排行榜

1. Linux下安装jdk8步骤详述(5)

```
[root@nginx-1 ~]#
[root@nginx-1 ~]# ls
keepalived-1.2.20.tar.gz  tengine-2.1.2.tar.gz
[root@nginx-1 ~]#
[root@nginx-1 ~]#
```

- 5、解包、编译安装

```
tar xzf keepalived-1.2.20.tar.gz -C /usr/local/src/
cd /usr/local/src/keepalived-1.2.20/
./configure --prefix=/usr/local/keepalived --with-kernel-dir=/usr/src/kernels/2.6.32-573.22.1.el6.x86_64/
make && make install
```

```
Keepalived configuration
-----
Keepalived version      : 1.2.20
Compiler                 : gcc
Compiler flags           : -g -O2
Extra Lib                : -lssl -lcrypto -lcrypt
Use IPVS Framework       : Yes
IPVS sync daemon support : Yes
IPVS use libnl           : No
fwmark socket support    : Yes
Use VRRP Framework       : Yes
Use VRRP VMAC            : Yes
Use VRRP authentication  : Yes
SNMP keepalived support  : No
SNMP checker support     : No
SNMP RFCv2 support       : No
SNMP RFCv3 support       : No
SHA1 support             : No
Use Debug flags          : No
libnl version            : None
Use IPv4 devconf         : No
Use libiptc              : No
Use libipset             : No
[root@nginx-1 keepalived-1.2.20]#
```

- 6、拷贝配置文件

```
mkdir -p /etc/keepalived
cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/
cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
cp /usr/local/keepalived/sbin/keepalived /usr/sbin/
```

- 7、创建脚本文件及目录（脚本我会在尾部添加）

三个脚本，分别为 keepalived 脚本脚本、nginx 检测脚本、nginx 同步脚本

```
mkdir -p /data/sh

touch /data/sh/keepalived_check.sh
touch /data/sh/nginx_check.sh
touch /data/sh/nginx_rsync.sh

chmod +x /data/sh/keepalived_check.sh
chmod +x /data/sh/nginx_check.sh
chmod +x /data/sh/nginx_rsync.sh
```

- 8、安装 mailx 包，用于发送邮件
- yum -y install mailx

编辑 /etc/mail.rc 文件，在尾部添加两行：

```
set from=xxx@163.com smtp=smtp.163.com
set smtp-auth-user=xxx@163.com smtp-auth-password=xxx smtp-auth=login
```

2. 更改MySQL数据库的编码为utf8mb4(5)
3. ActiveMQ实战-集群(3)
4. MyBatis的几种批量操作(2)
5. java 调用shell命令(2)

#### 9. keepalived 配置文件：



! Configuration File for keepalived

##### 全局配置 #####

global\_defs {

# 定义管理员邮件地址,表示keepalived在发生诸如切换操作时需要发送email通知,以及email发送给哪些邮件地址,可以有多个,每行一个

```
    notification_email {
        hongxue@showjoy.com
    }
```

# 表示发送通知的邮件源地址是谁

```
    notification_email_from keepalived@showjoy.com
```

# smtp服务器配置

```
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
```

# router\_id 则为机器标识

```
    router_id keepalived_1
```

# vrrp\_skip\_check\_adv\_addr

# vrrp\_strict

}

##### VRRPD配置 #####

# 定义nginx\_check脚本,脚本执行间隔10秒,权重10

```
#vrrp_script nginx_check {
#    script "/data/sh/nginx_check.sh"
#    interval 10
#    weight 10
#}
```

# 定义vrrp实例

```
vrrp_instance http {
```

# state定义instance的初始状态

```
    state BACKUP
```

# 实际绑定的网卡,配置虚拟IP的时候必须是在已有的网卡上面添加的

```
    interface eth0
```

# 要检测状态的网卡,当其中任意一块出现故障时keepalived都视为故障

```
    track_interface {
        eth0
#        eth1
    }
```

# 当该keepalived切换为MASTER状态时,执行下面的脚本

```
    notify_master /data/sh/keepalived_check.sh
```

# 这里设置VRID,相同的VRID为一个组,他将决定多播的MAC地址

```
    virtual_router_id 51
```

# 设置本节点优先级,高的为MASTER,如果优先级一样,则IP地址大的是MASTER

```
    priority 200
```

# 组播信息发送间隔,两个节点设置必须一样

```
    advert_int 10
```

# 验证方式与验证密码

```
    authentication {
        auth_type PASS
```

## 推荐排行榜

1. Linux下安装jdk8步骤详述(12)
2. 更改MySQL数据库的编码为utf8mb4(6)
3. maven依赖包下载失败解决办法(4)
4. Eclipse中.setting目录下文件介绍(4)
5. java.lang.ClassNotFoundException: org.springframework.web.util.IntrospectorCleanerListener(3)

```

    auth_pass 1111
}

# 这里设置VIP，它随着state变化而增加删除，当state为master的时候就添加，当state为backup的时候则删除，
# 由优先级决定，可以设置多个地址
virtual_ipaddress {
    192.168.1.60
    192.168.1.61
}

# 设置不抢占，这里只能设置在state为backup的节点上，而且这个节点的优先级必须别另外的高
#    noreempt

# 执行nginx检测脚本
#    track_script {
#        nginx_check weight 10
#    }
}

```



## 10、nginx 配置

### 安装nginx依赖

```
yum -y install pcre-devel pcre openssl-devel openssl
```

### 解包，编译安装

```
tar xzf tengine-2.1.2.tar.gz -C /usr/local/src/
```

```
cd /usr/local/src/tengine-2.1.2/
```

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

```
make && make install
```

## 11、启动nginx、启动脚本分别为：nginx\_check.sh nginx\_rsync.sh（可以看到，nginx 和 两个脚本都已经启动）

```

[root@nginx-1 sh]#
[root@nginx-1 sh]# ps -ef |grep nginx
root      12287      1   0 21:15 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    12288    12287   0 21:15 ?        00:00:00 nginx: worker process
root      12297      7432   0 21:16 pts/0    00:00:00 sh nginx_check.sh
root      12318      7432   0 21:16 pts/0    00:00:00 sh nginx_rsync.sh
root      12319    12318   0 21:16 pts/0    00:00:00 inotifywait -mrq -e modify,create,attrib,move,delete /usr/local/nginx/
root      12320    12318   0 21:16 pts/0    00:00:00 sh nginx_rsync.sh
root      12329      7432   0 21:16 pts/0    00:00:00 grep nginx
[root@nginx-1 sh]#

```

## 12、启动 keepalived

```

[root@nginx-1 log]#
[root@nginx-1 log]# /etc/init.d/keepalived start
Starting keepalived:
[root@nginx-1 log]#
[root@nginx-1 log]#
[root@nginx-1 log]# /etc/init.d/keepalived status
keepalived (pid 12312) is running...
[root@nginx-1 log]#

```

## 13、查看日志（可以看到，keepalived已经启动，并且成功绑定了 VIP 192.168.1.60和61）

```

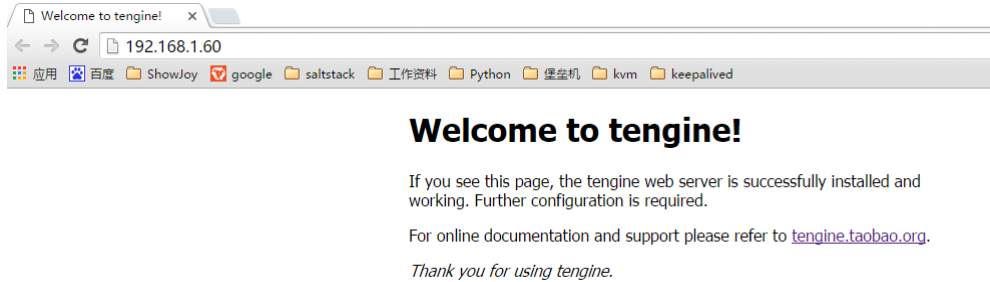
[root@nginx-1 log]# cat messages
Apr 17 21:19:42 nginx-1 Keepalived[12505]: Starting Keepalived v1.2.20 (04/17,2016)
Apr 17 21:19:42 nginx-1 Keepalived[12506]: Starting Healthcheck child process, pid=12507
Apr 17 21:19:42 nginx-1 Keepalived[12506]: Starting VRRP child process, pid=12508
Apr 17 21:19:42 nginx-1 Keepalived_healthcheckers[12507]: Netlink reflector reports IP 192.168.1.50 added
Apr 17 21:19:42 nginx-1 Keepalived_healthcheckers[12507]: Netlink reflector reports IP fe80::20c:29ff:fe38:5919 added
Apr 17 21:19:42 nginx-1 Keepalived_healthcheckers[12507]: Registering Kernel netlink reflector
Apr 17 21:19:42 nginx-1 Keepalived_healthcheckers[12507]: Registering Kernel netlink command channel
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Netlink reflector reports IP 192.168.1.50 added
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Netlink reflector reports IP fe80::20c:29ff:fe38:5919 added
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Registering Kernel netlink reflector
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Registering Kernel netlink command channel
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Registering gratuitous ARP shared channel
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) removing protocol VIPs.
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: Using LinkWatch kernel netlink reflector...
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) Entering BACKUP STATE
Apr 17 21:19:42 nginx-1 Keepalived_vrrp[12508]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]
Apr 17 21:19:42 nginx-1 Keepalived_healthcheckers[12507]: Using LinkWatch kernel netlink reflector...
Apr 17 21:20:13 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) Transition to MASTER STATE
Apr 17 21:20:23 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) Entering MASTER STATE
Apr 17 21:20:23 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) setting protocol VIPs.
Apr 17 21:20:23 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) Sending gratuitous ARPs on eth0 for 192.168.1.60
Apr 17 21:20:23 nginx-1 Keepalived_vrrp[12508]: VRRP_Instance(http) Sending gratuitous ARPs on eth0 for 192.168.1.61
Apr 17 21:20:23 nginx-1 Keepalived_healthcheckers[12507]: Netlink reflector reports IP 192.168.1.60 added
Apr 17 21:20:23 nginx-1 Keepalived_healthcheckers[12507]: Netlink reflector reports IP 192.168.1.61 added
[root@nginx-1 log]#
[root@nginx-1 log]#

```

14、查看网卡 IP 地址（除了自身IP地址以外、VIP也已经绑定）

```
[root@nginx-1 log]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:38:59:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global eth0
    inet 192.168.1.60/32 scope global eth0
    inet 192.168.1.61/32 scope global eth0
    inet6 fe80::20c:29ff:fe38:5919/64 scope link
        valid_lft forever preferred_lft forever
[root@nginx-1 log]#
[root@nginx-1 log]#
[root@nginx-1 log]#
```

15、通过VIP访问nginx



## 二、nginx-2（slave 主机配置）

1、前1-8步骤 nginx-1 和 nginx-2 是一样的，照着以前做就行，这里主要帖下 nginx-2 的 keepalived.conf 配置

2、keepalived 配置文件

```
! Configuration File for keepalived

##### 全局配置 #####

global_defs {

# 定义管理员邮件地址,表示keepalived在发生诸如切换操作时需要发送邮件通知,以及email发送给哪些邮件地址,
可以有多个,每行一个
    notification_email {
        hongxue@showjoy.com
    }

# 表示发送通知的邮件源地址是谁
    notification_email_from keepalived@showjoy.com

# smtp服务器配置
    smtp_server 127.0.0.1
    smtp_connect_timeout 30

# router_id 则为机器标识
    router_id keepalived_1

# vrrp_skip_check_adv_addr
# vrrp_strict

}

##### VRRPD配置 #####

# 定义vrrp脚本,脚本执行间隔10秒,权重10
#vrrp_script nginx_check {
#    script "/data/sh/nginx_check.sh"
#    interval 10
#    weight 10
#}
```

```

# 定义vrrp实例
vrrp_instance http {

# state定义instance的初始状态
state BACKUP

# 实际绑定的网卡，配置虚拟IP的时候必须是在已有的网卡上面添加的
interface eth0

# 要检测状态的网卡，当其中任意一块出现故障时keepalived都视为故障
track_interface {
    eth0
#    eth1
}

# 当该keepalived切换为MASTER状态时，执行下面的脚本
notify_master /data/sh/keepalived_check.sh

# 这里设置VRID，相同的VRID为一个组，他将决定多播的MAC地址
virtual_router_id 51

# 设置本节点优先级，高的为MASTER，如果优先级一样，则IP地址大的是MASTER
priority 200

# 组播信息发送间隔，两个节点设置必须一样
advert_int 10

# 验证方式与验证密码
authentication {
    auth_type PASS
    auth_pass 1111
}

# 这里设置VIP，它随着state变化而增加删除，当state为master的时候就添加，当state为backup的时候则删除，
由优先级决定，可以设置多个地址
virtual_ipaddress {
    192.168.1.60
    192.168.1.61
}

# 设置不抢占，这里只能设置在state为backup的节点上，而且这个节点的优先级必须别另外的高
#    nopreempt

# 执行nginx_check叫测脚本
#    track_script {
#        nginx_check weight 10
#    }
}

```

### 3、nginx 配置

安装nginx依赖

```
yum -y install pcre-devel pcre openssl-devel openssl
```

解包，编译安装

```
tar xzf engine-2.1.2.tar.gz -C /usr/local/src/
cd /usr/local/src/engine-2.1.2/
```

```
./configure --prefix=/usr/local/nginx
make && make install
```

4、启动nginx、启动 nginx\_check.sh (nginx\_rsync.sh 不必启动，因为只需配置一个nginx，则另一个也会自动同步并且 reload)

```

[root@nginx-2 sh]#
[root@nginx-2 sh]# ps -ef |grep nginx
root      12289      1   0 21:37 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    12290    12289   0 21:37 ?        00:00:00 nginx: worker process
root      12298    7420   0 21:38 pts/0    00:00:00 sh nginx_check.sh
root      12320    7420   0 21:39 pts/0    00:00:00 grep nginx

```

5、启动 keepalived 并查看日志

```
Apr 17 21:38:59 nginx-2 Keepalived[12312]: Starting Keepalived v1.2.20 (04/17/2016)
Apr 17 21:38:59 nginx-2 Keepalived[12313]: Starting Healthcheck child process, pid=12314
Apr 17 21:38:59 nginx-2 Keepalived[12313]: Starting VRRP child process, pid=12315
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Netlink reflector reports IP 192.168.1.51 added
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Netlink reflector reports IP fe80::20c:29ff:fe43:a57f added
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Registering Kernel netlink reflector
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Registering Kernel netlink command channel
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Registering gratuitous ARP shared channel
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 17 21:38:59 nginx-2 kernel: IPVS: Registered protocols (TCP, UDP, SCTP, AH, ESP)
Apr 17 21:38:59 nginx-2 kernel: IPVS: Connection hash table configured (size=4096, memory=64Kbytes)
Apr 17 21:38:59 nginx-2 kernel: IPVS: ipvs loaded.
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Netlink reflector reports IP 192.168.1.51 added
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Netlink reflector reports IP fe80::20c:29ff:fe43:a57f added
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Registering Kernel netlink reflector
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Registering Kernel netlink command channel
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: VRRP_Instance(http) removing protocol VIPs.
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: Using LinkWatch kernel netlink reflector...
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: VRRP_Instance(http) Entering BACKUP STATE
Apr 17 21:38:59 nginx-2 Keepalived_vrrp[12315]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]
Apr 17 21:38:59 nginx-2 Keepalived_healthcheckers[12314]: Using LinkWatch kernel netlink reflector...
[root@nginx-2 log]#
```

### 三、故障及需求演练

#### 1、当 keepalived 挂掉，那么理论上说 keepalived 服务会重新启动起来（通过nginx\_check.sh 脚本实现）

在下图中可以看到，我第一次 status 之后 keepalived 的状态是 running 的，当我执行 stop 之后 keepalived 马上变为 stopped 状态，可过会之后 keepalived 状态又变为 running

```
[root@nginx-1 log]#
[root@nginx-1 log]# /etc/init.d/keepalived status
keepalived (pid 13649) is running...
[root@nginx-1 log]# /etc/init.d/keepalived stop
Stopping keepalived: [ OK ]
[root@nginx-1 log]# /etc/init.d/keepalived status
keepalived is stopped
[root@nginx-1 log]# /etc/init.d/keepalived status
keepalived (pid 13709) is running...
[root@nginx-1 log]#
```

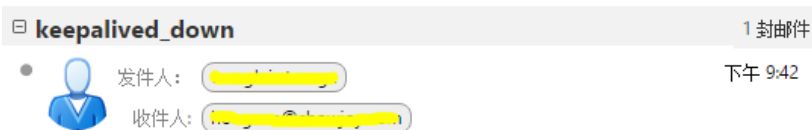
#### 2、当 nginx 挂掉后，那么理论上说 keepalived 服务也会关闭（通过nginx\_check.sh 脚本实现）

在下图中可以看到，第一次我 status 的时候 keepalived 的状态是 running... 的，nginx 进程也在，当我 killall 掉 nginx 之后，前三秒 keepalived 的状态还是 running... 接着马上停止该服务（脚本检测时间为10秒一次）

```
[root@nginx-2 ~]#
[root@nginx-2 ~]# service keepalived status
keepalived (pid 13340) is running...
[root@nginx-2 ~]#
[root@nginx-2 ~]# ps -ef |grep nginx
root      12298      7420    0 21:38 pts/0    00:00:00 sh nginx_check.sh
root      13324        1    0 21:59 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    13325    13324    0 21:59 ?        00:00:00 nginx: worker process
root      13366      7420    0 21:59 pts/0    00:00:00 grep nginx
[root@nginx-2 ~]#
[root@nginx-2 ~]#
[root@nginx-2 ~]# killall nginx
[root@nginx-2 ~]# service keepalived status
keepalived (pid 13340) is running...
[root@nginx-2 ~]# service keepalived status
keepalived (pid 13340) is running...
[root@nginx-2 ~]# service keepalived status
keepalived (pid 13340) is running...
[root@nginx-2 ~]# service keepalived status
keepalived dead but subsys locked
[root@nginx-2 ~]# service keepalived status
keepalived dead but subsys locked
[root@nginx-2 ~]#
```

#### 3、当 keepalived 状态变为 Master 之后，发送邮件（通过 keepalived\_check.sh 脚本实现）

在第一个问题中，我停止 keepalived 服务，那么这时候 slave 就会将状态转换为 master，则相应的，我也会收到报警邮件或者短信



“2016-04-17-21-42 keepalived-1 is a down,keepalived-2 is a MASTER”







```
#!/bin/bash

while :
do

# 获取nginx、keepalived的进程数
nginxcheck=`ps -C nginx --no-header | wc -l`
keepalivedcheck=`ps -C keepalived --no-header | wc -l`

# 如果检测到(nginx)进程数等于0, 并且(keepalived)进程数不等于0, 则关闭(keepalived)
if [ $nginxcheck -eq 0 ];then
    if [ $keepalivedcheck -ne 0 ];then
        killall -TERM keepalived
    else
        echo "keepalived is stoped"
    fi
fi

# 如果检测到(nginx)进程数不等于0, 则(keepalived)进程数等于0, 那么启动keepalived服务
else
    if [ $keepalivedcheck -eq 0 ];then
        /etc/init.d/keepalived start
    else
        echo "keepalived is running"
    fi
fi

sleep 10
done
```



### 3、nginx\_rsync.sh 脚本（执行方式：后台，nohup sh nginx\_rsync.sh 该脚本只有一只主机在后台执行）



```
#!/bin/bash

ip_address="192.168.1.51"

inotify_cmd="inotifywait -mrq -e modify,create,attrib,move,delete /usr/local/nginx/"
rsync_cmd="rsync -azH --delete /usr/local/nginx/ root@${ip_address}:/usr/local/nginx/"

$inotify_cmd | while read DIRECTORY EVENT FILE
do
    if [ $(pgrep rsync | wc -l) -le 0 ];then
        $rsync_cmd
        ssh root@${ip_address} "/usr/local/nginx/sbin/nginx -t"
        if [ $? -eq 0 ];then
            ssh root@${ip_address} "/usr/local/nginx/sbin/nginx -s reload"
        fi
    fi
done
```



## keepalived nginx 双机热备实战精讲

在试验之前先贴张图上来，大致说下具体的需求及环境（注：实验所用到的脚本在文章末尾，如果在过程中遇到可以先看脚本）

### 环境：

VIP（漂移地址）：192.168.1.60-192.168.1.61

nginx-1：192.168.1.50

nginx-2：192.168.1.51

关闭 iptables、关闭selinux 并且打通ssh配置，使得192.168.1.50和192.168.1.51可以免密码登录（主要同于同步 nginx 配置和 reload，公钥验证文件：authorized\_keys）

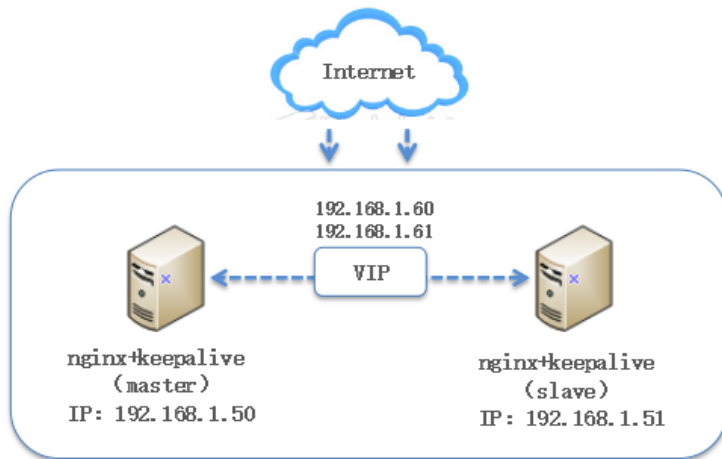
### 需求：

两只 nginx web 服务器分别提供相同的访问，当一只主机无法提供服务（keepalived、nginx 或者说主机宕机）后，另一只马上转换为MASTER状态，并且继承VIP，继续向用户提供服务。

- 1、如果 keepalived 状态转换为 MASTER，则发送邮件进行通知运维人员（由下面 keepalived\_check.sh 脚本实现）
- 2、需要nginx两台服务器提供的服务是完全一致的，并且当一台 nginx 的配置改变，则另一台立刻变为相同的配置，

并且 reload。（由 rsync+inotify 实时同步脚本 nginx\_rsync.sh 脚本实现）

3、如果keepalived 挂掉，则检测并且重启keepalived；如果nginx 挂掉，则关闭keepalived，且让VIP转移到 keepalived-slave上面（由 nginx\_check.sh脚本实现）



下面开始实验：

## 一、nginx-1（master 主机配置）

1、时间校准

```
ntpdate ntp.api.bz
```

2、安装依赖软件

```
yum -y install openssl openssl-devel kernel-devel
```

3、使用inotify机制，还需要安装inotify-tools工具，以便提供inotifywait,inotifywatch辅助工具，用来监控，汇总改动情况

```
yum -y install inotify-tools
```

4、准备 keepalived 和 nginx 包（keepalived 包可以去 keepalived.org 下载，nginx 包则去 tengine 官方下载）

```
[root@nginx-1 ~]#  
[root@nginx-1 ~]# ls  
keepalived-1.2.20.tar.gz  tengine-2.1.2.tar.gz  
[root@nginx-1 ~]#  
[root@nginx-1 ~]#
```

5、解包、编译安装

```
tar zxf keepalived-1.2.20.tar.gz -C /usr/local/src/  
cd /usr/local/src/keepalived-1.2.20/  
./configure --prefix=/usr/local/keepalived --with-kernel-dir=/usr/src/kernels/2.6.32-573.  
make && make install
```

```
Keepalived configuration
-----
Keepalived version      : 1.2.20
Compiler                : gcc
Compiler flags          : -g -O2
Extra Lib               : -lssl -lcrypto -lcrypt
Use IPVS Framework      : Yes
IPVS sync daemon support : Yes
IPVS use libnl          : No
fwmark socket support   : Yes
Use VRRP Framework      : Yes
Use VRRP VMAC           : Yes
Use VRRP authentication : Yes
SNMP keepalived support : No
SNMP checker support     : No
SNMP RFCv2 support       : No
SNMP RFCv3 support       : No
SHA1 support            : No
Use Debug flags         : No
libnl version           : None
Use IPv4 devconf         : No
Use libiptc              : No
Use libipset             : No
[root@nginx-1 keepalived-1.2.20]#
```

搜索微信号：a168888vip

如果我写的东西帮到了您，请  
大佬您打开支付宝扫一下二维  
码领个红包吧！（记得用掉红  
包啊，如果不用支付宝买东西，这个红包也不要浪费呀，  
使用支付宝施舍给我也好，[点这里我用掉](#)）

支付宝ALIPAY

扫码领红包

天天可领 想花就花

打开支付宝[扫一扫]

支付就用支付宝

大自然的搬运工(\*海明)

打开支付宝[扫一扫]

免费寄送收钱码：拨打95188-6

推荐使用微信支付

大自然的搬运工(\*\*明)

微信支付

好文置顶 关注我 收藏该文

这个名字想了很久~

关注 - 8

粉丝 - 167

+加关注

00

« 上一篇：[java IO流之三 使用缓冲流来读写文件](#)

» 下一篇：[Ngrok 内网穿透利器](#)

posted @ 2017-05-17 16:24

这个名字想了很久~

阅读(1682)

评论(1)

编辑

收藏

评论

#1楼 2018-09-17 16:40 | 依然是等待

怎么查看主/备的状态，命令是什么？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万C++/C#源码: 大型实时仿真HMI组态CAD\GIS图形源码！
- 【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

最新新闻：

- [AirPods是如何像病毒一样流行起来的？](#)
  - [肥宅快乐水越来越不好卖，可口可乐和百事可乐都下调了销售预期](#)
  - [关于极限分类，这里有你想知道的一切](#)
  - [门捷列夫的化学元素周期表发表150周年](#)
  - [机器学习加速“科学危机”](#)
- » [更多新闻...](#)

Copyright ©2019 这个名字想了很久~