

## Keepalived之——Keepalived + Nginx 实现高可用 Web 负载均衡

### 一、Keepalived 简要介绍

Keepalived 是一种高性能的服务器高可用或热备解决方案，Keepalived 可以用来防止服务器单点故障的发生，通过配合 Nginx 可以实现 web 前端服务的高可用。

Keepalived 以 VRRP 协议为实现基础，用 VRRP 协议来实现高可用性(HA)。VRRP(Virtual RouterRedundancy Protocol)协议是用于实现路由器冗余的协议，VRRP 协议将两台或多台路由器设备虚拟成一个设备，对外提供虚拟路由器 IP(一个或多个)，而在路由器组内部，如果实际拥有这个对外 IP 的路由器如果工作正常的话就是 MASTER，或者是通过算法选举产生，MASTER 实现针对虚拟路由器 IP 的各种网络功能，如 ARP 请求，ICMP，以及数据的转发等；其他设备不拥有该虚拟 IP，状态是 BACKUP，除了接收 MASTER 的VRRP 状态通告信息外，不执行对外的网络功能。当主机失效时，BACKUP 将接管原先 MASTER 的网络功能。VRRP 协议使用多播数据来传输 VRRP 数据，VRRP 数据使用特殊的虚拟源 MAC 地址发送数据而不是自身网卡的 MAC 地址，VRRP 运行时只有 MASTER 路由器定时发送 VRRP 通告信息，表示 MASTER 工作正常以及虚拟路由器 IP(组)，BACKUP 只接收 VRRP 数据，不发送数据，如果一定时间内没有接收到 MASTER 的通告信息，各 BACKUP 将宣告自己成为 MASTER，发送通告信息，重新进行 MASTER 选举状态。

### 二、方案规划

VIP IP 主机名 Nginx端口 默认主从

192.168.50.130 192.168.50.133 liuyazhuang133 88 MASTER

192.168.50.130 192.168.50.134 liuyazhuang134 88 BACKUP

CentOS 6.5 x64

keepalived-1.2.18.tar.gz

nginx-1.13.0.tar.g

### 三、安装Nginx

#### 1、参考《Nginx+Tomcat+Memcached负载均衡集群服务搭建》

大家可以参考博文《Nginx+Tomcat+Memcached负载均衡集群服务搭建》中安装Nginx部分进行Nginx的安装，也可以按照如下步骤安装Nginx。

#### 2、安装编译 Nginx 所需的依赖包

```
# yum install gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel  
openssl openssl-devel
```

#### 3、下载Nginx

```
# cd /usr/local/src/  
# wget http://nginx.org/download/nginx-1.13.0.tar.gz
```

#### 4、编译安装 Nginx

```
# cd /usr/local/src/  
# tar -zxvf nginx-1.13.0.tar.gz  
# cd nginx-1.13.0  
# ./configure --prefix=/usr/local/nginx  
# make && make install
```

#### 5、配置 Nginx

```
# vi /usr/local/nginx/conf/nginx.conf  
  
user root;  
worker_processes 1;  
#error_log logs/error.log;  
#error_log logs/error.log notice;  
#error_log logs/error.log info;  
#pid logs/nginx.pid;  
events {  
worker_connections 1024;  
}  
http {  
include mime.types;  
default_type application/octet-stream;  
#log_format main '$remote_addr - $remote_user [$time_local] "$request" '  
# '$status $body_bytes_sent "$http_referer" '  
# '"$http_user_agent" "$http_x_forwarded_for";  
#access_log logs/access.log main;  
sendfile on;  
#tcp_nopush on;  
#keepalive_timeout 0;  
keepalive_timeout 65;  
#gzip on;  
server {
```

```
listen 88;
server_name localhost;
#charset koi8-r;
#access_log logs/host.access.log main;
location / {
root html;
index index.html index.htm;
}
#error_page 404 /404.html;
# redirect server error pages to the static page /50x.html
error_page 500 502 503 504 /50x.html;
location = /50x.html {
root html;
}
}
```

修改 Nginx 欢迎首页内容（用于后面测试， 用于区分两个节点的 Nginx）：

```
# vi /usr/local/nginx/html/index.html
192.168.50.133 中的标题加 1
<h1>Welcome to nginx! 1</h1>
192.168.50.134 中的标题加 2
<h1>Welcome to nginx! 2</h1>
```

## 6、系统防火墙打开对应的端口 88

```
# vi /etc/sysconfig/iptables
## Nginx
-A INPUT -m state --state NEW -m tcp -p tcp --dport 88 -j ACCEPT
# service iptables restart
```

## 7、测试 Nginx 是否安装成功

```
# /usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

## 8、启动 Nginx

```
# /usr/local/nginx/sbin/nginx
```

重启 Nginx

```
# /usr/local/nginx/sbin/nginx -s reload
```

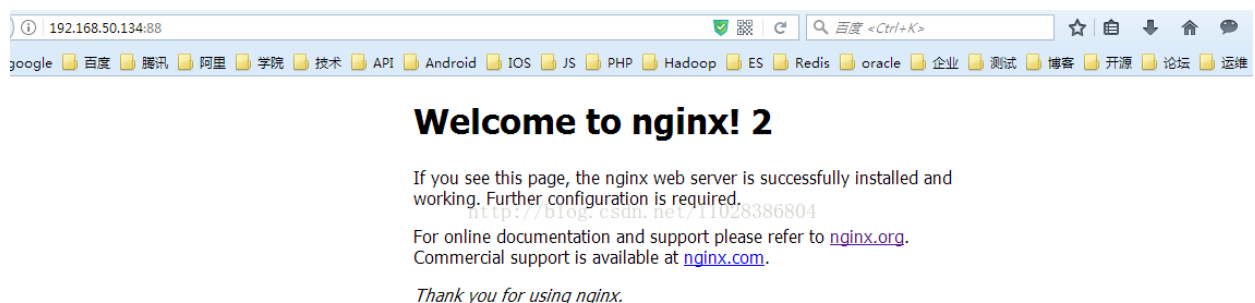
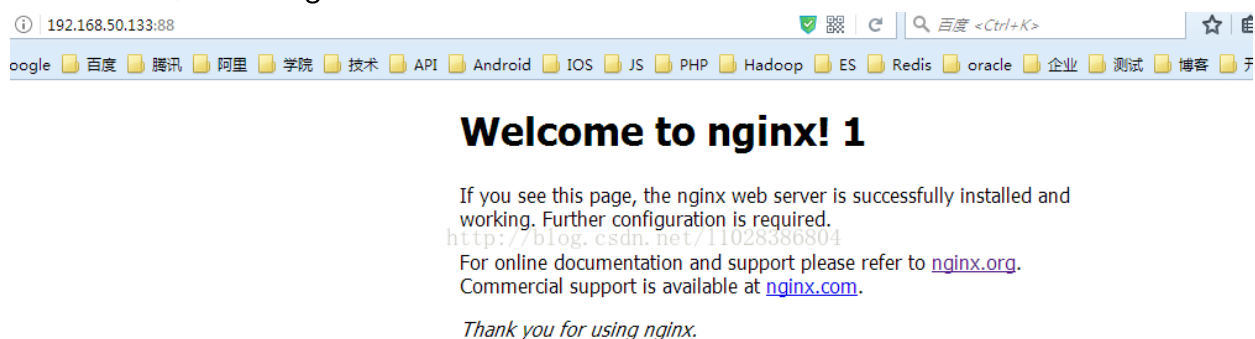
## 9、设置 Nginx 开机启动

```
# vi /etc/rc.local
```

加入：

```
/usr/local/nginx/sbin/nginx
```

## 10、分别访问两个 Nginx



## 四、安装 Keepalived

(<http://www.keepalived.org/download.html> )

## 1、 上传或下载 keepalived

上传或下载 keepalived (keepalived-1.2.18.tar.gz) 到 /usr/local/src 目录

## 2、 解压安装

```
# cd /usr/local/src
# tar -zxvf keepalived-1.2.18.tar.gz
# cd keepalived-1.2.18
# ./configure --prefix=/usr/local/keepalived
# make && make install
```

## 3、 将 keepalived 安装成 Linux 系统服务

因为没有使用 keepalived 的默认路径安装（默认是/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

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

复制 keepalived 服务脚本到默认的地址

```
# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
# cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
# ln -s /usr/local/sbin/keepalived /usr/sbin/
# ln -s /usr/local/keepalived/sbin/keepalived /sbin/
```

设置 keepalived 服务开机启动

```
# chkconfig keepalived on
```

## 4、 修改 Keepalived 配置文件

(1) MASTER 节点配置文件 (192.168.50.133)

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived
```

```
global_defs {
## keepalived 自带的邮件提醒需要开启 sendmail 服务。 建议用独立的监控或第三方 SMTP
router_id liuyazhuang133 ## 标识本节点的字条串，通常为 hostname
}

## keepalived 会定时执行脚本并对脚本执行的结果进行分析，动态调整 vrrp_instance 的优先级。如果脚本执行结果为 0，并且 weight 配置的值大于 0，则优先级相应的增加。如果脚本执行结果非 0，并且 weight配置的值小于 0，则优先级相应的减少。其他情况，维持原本配置的优先级，即配置文件中 priority 对应的值。

vrrp_script chk_nginx {
script "/etc/keepalived/nginx_check.sh" ## 检测 nginx 状态的脚本路径
interval 2 ## 检测时间间隔
weight -20 ## 如果条件成立，权重-20
}

## 定义虚拟路由， VI_1 为虚拟路由的标示符，自己定义名称
vrrp_instance VI_1 {
state MASTER ## 主节点为 MASTER， 对应的备份节点为 BACKUP
interface eth0 ## 绑定虚拟 IP 的网络接口，与本机 IP 地址所在的网络接口相同， 我的是 eth0
virtual_router_id 33 ## 虚拟路由的 ID 号， 两个节点设置必须一样， 可选 IP 最后一段使用，相同的 VRID 为一个组，他将决定多播的 MAC 地址
mcast_src_ip 192.168.50.133 ## 本机 IP 地址
priority 100 ## 节点优先级， 值范围 0-254， MASTER 要比 BACKUP 高
nopreempt ## 优先级高的设置 nopreempt 解决异常恢复后再次抢占的问题
advert_int 1 ## 组播信息发送间隔，两个节点设置必须一样， 默认 1s
## 设置验证信息，两个节点必须一致
authentication {
auth_type PASS
auth_pass 1111 ## 真实生产，按需求对应该过来
}

## 将 track_script 块加入 instance 配置块
track_script {
chk_nginx ## 执行 Nginx 监控的服务
} #

# 虚拟 IP 池, 两个节点设置必须一样
virtual_ipaddress {
192.168.50.130 ## 虚拟 ip，可以定义多个
}
}
```

## (2)BACKUP 节点配置文件 (192.168.50.134)

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
router_id liuyazhuang134
}
vrrp_script chk_nginx {
script "/etc/keepalived/nginx_check.sh"
interval 2
weight -20
}
vrrp_instance VI_1 {
state BACKUP
interface eth1
virtual_router_id 33
mcast_src_ip 192.168.50.134
priority 90
advert_int 1
authentication {
auth_type PASS
auth_pass 1111
}
track_script {
chk_nginx
}
virtual_ipaddress {
192.168.50.130
}
}
```

## 5、编写 Nginx 状态检测脚本

编写 Nginx 状态检测脚本 /etc/keepalived/nginx\_check.sh (已在 keepalived.conf 中配置)脚本要求：如果 nginx 停止运行，尝试启动，如果无法启动则杀死本机的 keepalived 进程，keepalived将虚拟 ip 绑定到 BACKUP 机器上。内容如下：

```
# vi /etc/keepalived/nginx_check.sh
#!/bin/bash
A=`ps -C nginx --no-header |wc -l`
if [ $A -eq 0 ];then
/usr/local/nginx/sbin/nginx
sleep 2
if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then
killall keepalived
fi
fi
```

保存后，给脚本赋执行权限：

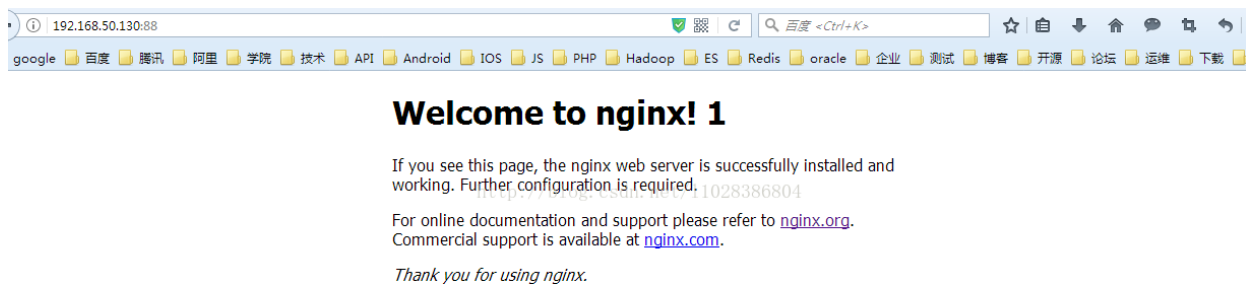
```
# chmod +x /etc/keepalived/nginx_check.sh
```

## 6、启动 Keepalived

```
# service keepalived start
Starting keepalived: [ OK ]
```

## 7、Keepalived+Nginx 的高可用测试

同时启动192.168.50.133和192.168.50.134上的Nginx和Keepalived，我们通过VIP(192.168.50.130)来访问Nginx,如下



我们关闭192.168.50.133上的Keepalived和Nginx

在192.168.50.133执行如下命令：

```
service keepalived stop
```



```
/usr/local/nginx/sbin/nginx -s stop
```

此时，再通过VIP(192.168.50.130)来访问Nginx,如下



我们再开启192.168.50.133上的Keepalived和Nginx

在192.168.50.133执行如下命令：

```
/usr/local/nginx/sbin/nginx  
service keepalived start
```

或者只执行

```
service keepalived start
```

因为我们写了脚本 `nginx_check.sh`，这个脚本会为我们自动启动Nginx。

此时，我们再通过VIP(192.168.50.130)来访问Nginx,如下

至此，Keepalived + Nginx 实现高可用 Web 负载均衡搭建完毕。