

高可用Nginx集群安装搭建手册

LVS搭建Nginx集群

准备工作

环境说明

共需要三台linux centos服务器，一台LVS，两台RealServer，**端口号必须保持一致，设为80**，所以需要3台服务器。

设定IP环境如下

| 服务名 | IP | 端口 | 作用 |
|-------------------|--|----|--------------------------|
| LVS-Director | VIP 192.168.120.200 RIP 192.168.120.58 | 80 | 运行LVS均衡调度，对外提供虚拟IP访问 |
| RealServer-Nginx1 | 192.168.120.83 | 80 | 运行nginx及tomcat服务，作为真实服务1 |
| RealServer-Nginx2 | 192.168.120.58 | 80 | 运行nginx及tomcat服务，作为真实服务2 |

LVS-Director负责将80端口的请求，负载均衡到Nginx1、Nginx2两台真实服务器上去，接下来我们将进行配置LVS-Director的路由方式-DR，直接路由。

准备IP地址信息打印程序balancer-1.0.0.jar，<http://hostname:port/server/ip>，打印服务器的IP端口号信息。

启动RealServer服务

安装步骤，参考前面Nginx的手册来操作，将Nginx分别安装在前面准备的两台linux服务器上。

启动Nginx，都为80端口，代理balancer-1.0.0.jar服务。

balancer-1.0.0.jar的端口号，分别为8081

操作192.168.120.83、192.168.120.58

```
java -jar balancer-1.0.0.jar --server.port=8081
cd /usr/local/nginx
sudo ./nginx -c conf/nginx_lvs_upstream.conf
```

nginx_lvs_upstream.conf配置内容，提供在课件资料中

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
```

```

include      mime.types;
default_type text/html;
sendfile     on;
keepalive_timeout 65;

upstream backend {
    server 127.0.0.1:8081;
}

server {
    location / {
        proxy_pass http://backend;
    }
}
}

```

测试各个端口的服务是否可用, <http://hostname:port/server/ip>

安装LVS

LVS全称为Linux Virtual Server, 工作在ISO模型中的第四层, 由于其工作在第四层, 因此与iptables类似, 必须工作在内核空间上。因此lvs与iptables一样, 是直接工作在内核中的, 叫ipvs, 主流的linux发行版默认都已经集成了ipvs, 因此用户只需安装一个管理工具ipvsadm即可。LVS现在已成为Linux内核的一部分, 默认编译为ip_vs模块, 必要时能够自动调用。以下操作可以手动加载ip_vs模块, 并查看当前系统中ip_vs模块的版本信息。

加载LVS

```

# 通过下面命令来检查, 如果没有显示, 则说明没有加载
lsmod |grep ip_vs
ip_vs                145497  0
# 加载LVS, 执行下面命令就可以把ip_vs模块加载到内核
# 或者如果是centos7, 通过ipvsadm来加载, 可以跳过这步
sudo modprobe ip_vs
# 查看LVS版本号, 说明安装成功
cat /proc/net/ip_vs
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port Forward Weight ActiveConn InActConn

```

IPVS管理工具ipvsadm

ipvsadm是lvs的管理工具, 我们在使用ipvs的时候需要使用到这个工具。

安装依赖

```

sudo yum install -y gcc gcc-c++ makepcr pcre-devel kernel-devel openssl-devel
libnl-devel popt*

```

安装ipvsadm

```

curl "http://www.linuxvirtualserver.org/software/kernel-2.6/ipvsadm-1.26.tar.gz"
-o ipvsadm-1.26.tar.gz
tar xzf ipvsadm-1.26.tar.gz
cd ipvsadm-1.26

```

```
rpm -qa | grep kernel-devel # 确认是否安装了kernel-devel（默认已经安装）
sudo make && sudo make install

curl "http://www.linuxvirtualserver.org/software/kernel-2.6/ipvsadm-1.25-1.src.rpm" -o ipvsadm-1.25-1.src.rpm
sudo rpm -ivh ipvsadm-1.25-1.src.rpm

# 检查是否安装成功，显示下面的内容表示安装成功
sudo ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward weight ActiveConn InActConn
```

LVS DR模式

DR模式是LVS三种实现负载均衡方式中性能最好的一个，下面就使用这个模式，配置Nginx负载均衡的实现。

LVS服务配置

编辑lvs文件

```
sudo vim /usr/local/sbin/lvs_dr.sh
```

输入以下内容

```
#!/bin/bash
# 1, 启用ip转发; 0, 禁止ip转发; 默认0。
echo 1 > /proc/sys/net/ipv4/ip_forward
# 定义两个变量方便后面使用
ipv=/sbin/ipvsadm
vip=192.168.120.200
# 通过ifconfig, 找到自己的网卡名, 我这里是enp0s3
# 下掉enp0s3:0的虚拟ip
ifconfig enp0s3:0 down
# 通过ifconfig, 找到自己的网卡名, 在其下面绑定虚拟ip
# 在enp0s3上绑定虚拟ip, 虚拟ip地址的广播地址是它本身
ifconfig enp0s3:0 $vip broadcast $vip netmask 255.255.255.255 up
# 添加路由规则
route add -host $vip dev enp0s3:0
# 清除原有转发规则
$ipv -C
# 新增虚拟服务端口, 采用轮询策略, 负载转发端口是80
$ipv -A -t $vip:80 -s rr

# 定义实际服务ip
rs1=192.168.120.103
rs2=192.168.120.58
# -a在虚拟IP中添加上游服务信息; -t表示tcp服务
# -r表示真实服务信息; -g指定为LVS为直接路由模式
$ipv -a -t $vip:80 -r $rs1:80 -g
$ipv -a -t $vip:80 -r $rs2:80 -g
$ipv --set 1 2 1
```

使配置生效

```
sudo sh /usr/local/sbin/lvs_dr.sh
sudo ipvsadm
# 查看虚拟服务端端口转发表
sudo ipvsadm -ln
```

可能出现的错误

0、SIOCSIFFLAGS：无法指定被请求的地址

没有对应的enp0s3:0网卡ip，本来就没有，可以忽略。运行脚本后enp0s3:0正常出现就没有问题。

1、没有轮询效果

一条tcp的连接经过lvs后，lvs会把这台记录保存15分钟

```
sudo ipvsadm -L --timeout # 查看timeout配置信息
```

可以设置短一点，达到实验效果：

```
sudo ipvsadm --set 1 2 1
```

注：保存添加的虚拟ip记录和ipvsadm的规则可以使用service ipvsadm save，还可以用-S或--save。清除所有记录和规则除了使用-C，还可以使用--clear

2、错误：Memory allocation problem

查看一下vmalloc使用情况：

```
cat /proc/meminfo | grep -i vmalloc
```

在/etc/default/grub文件的末尾添加如下行：

```
GRUB_CMDLINE_LINUX="vmalloc=256MB"
```

完成上述操作之后，发现lvs状态仍然是SYN_RECV。

3、lvs状态仍然是SYN_RECV

抓包后的pcap文件中，没有syn ack。于是想到是不是在什么地方丢掉了。

看到官方文档中有描述要设置re_filter。

查了一下这个参数的解释

```
=====
rp_filter参数有三个值，0、1、2，具体含义：
```

0：不开启源地址校验。

1：开启严格的反向路径校验。对每个进来的数据包，校验其反向路径是否是最佳路径。如果反向路径不是最佳路径，则直接丢弃该数据包。

2：开启松散的反向路径校验。对每个进来的数据包，校验其源地址是否可达，即反向路径是否能通（通过任意网口），如果反向路径不同，则直接丢弃该数据包。

```
=====
```

default的值是1，这里改为2

```
echo 2 > /proc/sys/net/ipv4/conf/网卡名/rp_filter
```

```
echo 2 > /proc/sys/net/ipv4/conf/网卡名/rp_filter
```

```
systemctl restart network.service
```

```
echo "1" > /proc/sys/net/ipv4/conf/lo/arp_ignore
```

```
echo "2" > /proc/sys/net/ipv4/conf/lo/arp_announce
```

```
echo "1" > /proc/sys/net/ipv4/conf/all/arp_ignore
```

```
echo "2" > /proc/sys/net/ipv4/conf/all/arp_announce
```

ipvsadm操作说明

```
sudo ipvsadm -C 清除
```

说明:

-A --add-service 在服务器列表中新添加一条新的虚拟服务器记录

-t 表示为tcp服务

-u 表示为udp服务

-s --scheduler 使用的调度算法, rr | wrr | lc | wlc | lb1b | lb1cr | dh | sh | sed | nq 默认调度算法是 wlc

-a --add-server 在服务器表中添加一条新的真实主机记录

-r --real-server 真实服务器地址

-m --masquerading 指定LVS工作模式为NAT模式

-w --weight 真实服务器的权值

-g --gatewaying 指定LVS工作模式为直接路由器模式 (也是LVS默认的模式)

-i --ipip 指定LVS的工作模式为隧道模式

sudo ipvsadm -help 可以查看更多的帮助信息

真实服务配置

在lvs的DR和TUN模式下, 用户的访问请求到达真实服务器后, 响应数据是直接返回给用户的, 而不再经过前端的Director Server, 因此, 就需要在每个Real server节点上增加虚拟的VIP地址, 这样数据才能直接返回给用户。

2台服务都需要配置相同的内容, 2台真实web服务器的配置信息, 编辑lvs_dr_rs.sh文件

```
sudo vim /usr/local/sbin/lvs_dr_rs.sh
```

输入内容

```
#!/bin/bash
vip=192.168.120.200
ifconfig enp0s3:0 $vip broadcast $vip netmask 255.255.255.255 up
# 使用回环网卡适配器
route add -host $vip lo:0
# 关闭arp解析
echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
```

关闭arp解析

arp_ignore: 当ARP请求发过来后发现自己正是请求的地址是否响应;

- 0 - 利用本地的任何地址, 不管配置在哪个接口上去响应ARP请求;
- 1 - 哪个接口上接受ARP请求, 就从哪个端口上回应。

arp_announce: 定义不同级别, 当ARP请求通过某个端口进来是否利用这个接口来回应。

- 0 - 利用本地的任何地址, 不管配置在哪个接口上去响应ARP请求;
- 1 - 避免使用另外一个接口上的mac地址去响应ARP请求;
- 2 - 尽可能使用能够匹配到ARP请求的最佳地址。

使配置生效

```
sudo sh /usr/local/sbin/lvs_dr_rs.sh
sudo sysctl -p
```

查看日志

LVS的日志是通过syslog来输出的，通过下面的命令就能查看最新日志输出，方便我们查看错误信息

```
sudo tail -f /var/log/messages
```

检查是否生效

多出enp0s3:0:虚拟网卡

```
ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.120.127 netmask 255.255.255.0 broadcast 192.168.120.255
    inet6 fe80::370a:a3ea:20d0:69e3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f1:e3:c4 txqueuelen 1000 (Ethernet)
    RX packets 123143 bytes 129065693 (123.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28035 bytes 2598163 (2.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 88 bytes 9145 (8.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
lo:0: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 192.168.100.200 netmask 255.255.255.255
    loop txqueuelen 1000 (Local Loopback)
```

可能出现的错误

不能telnet端口

现象描述 VIP能ping通，但是VIP对应的端口号telnet不通

解决方法

确认真实服务器VIP是否一致

真实服务器与LVS-Director的端口号必须相同

测试Nginx集群效果

在浏览器输入访问

```
http://192.168.120.200:8080/server/ip
```

如果你发现IP地址在变化，恭喜你，成功了！

封装成shell脚本

LVS-Director脚本

见课件资料《lvs_dr.sh》shell脚本，在LVS-DR机器上我们只需要执行lvs_dr.sh脚本即可

使用方法

```
sudo sh lvs_dr.sh --help
Usage: lvs_dr.sh {start | stop}
Usage: lvs_dr.sh start vip vip_port "real_server_ips..." virtual_net_card
Usage: lvs_dr.sh stop virtual_net_card

# 启动, 指定虚拟ip、真实服务ip、虚拟网卡
sudo sh lvs_dr.sh start 192.168.120.200 80 "192.168.120.83 192.168.120.58"
enp0s3:0
# 停止
sudo sh lvs_rs.sh stop enp0s3:0
```

RealServer脚本

见课件资料《lvs_rs.sh》shell脚本，在RealServer机器上我们只需要执行lvs_rs.sh脚本即可

使用方法

```
sh lvs_rs.sh --help
Usage: lvs_rs.sh {start | stop}
Usage: lvs_rs.sh start vip loopback
Usage: lvs_rs.sh stop loopback

# 启动, 指定虚拟ip、虚拟网卡
sudo sh lvs_rs.sh start 192.168.120.200 lo:0
# 停止
sudo sh lvs_rs.sh stop lo:0
```

window下编写的shell脚本，在linux下运行有 `$'\r': 未找到命令` 的错误，是因为编码格式问题，通过dos2unix来解决。

```
sudo yum install -y dos2unix
sudo dos2unix lvs_rs.sh
```

Keepalived保证应用高可用

搭建Nginx高可用

环境准备

共需要两台linux centos服务器，一台作为Nginx Master，一台作为Nginx Backup。两台服务器上都需要安装Keepalived，通过Keepalived来开启虚拟端口，以及保证Nginx高可用。

| 服务名 | IP | 端口 | 作用 |
|-----------------------------------|--|----|--------------------------|
| Nginx-Master Keepalived-Master | RIP 192.168.120.58 VIP 192.168.120.200 | 80 | 运行LVS均衡调度，对外提供虚拟IP访问 |
| Nginx-Backup Keepalived-Backup | RIP 192.168.120.83 VIP 192.168.120.200 | 80 | 运行nginx及tomcat服务，作为真实服务1 |

Keepalived不仅仅保证LVS高可用，也能保证其他应用高可用，比如tomcat、Nginx、甚至RPC服务。

我们需要**三份配置文件**（一个nginx_monitor监控脚本，主备LVS各一份keepalived配置）

Nginx监控脚本

在监控脚本中，通过ps命令来不停的探测nginx进程是否存在，来判断nginx是否还可用。编辑nginx_monitor.sh文件

```
sudo vim /etc/keepalived/nginx_monitor.sh
```

脚本内容

```
#!/bin/bash
# shell脚本监控：如果程序的进程存在，则认为没有问题
if [ "$(ps -ef | grep "nginx"| grep -v grep| grep -v keepalived )" == "" ]
then
    # 输出日志到系统日志中
    echo "nginx ##### app down " >> /var/log/messages
    exit 1
else
    # 一切正常
    exit 0
fi
```

执行脚本

```
# 创建nginx_monitor 脚本，并赋予可执行权限
sudo chmod +x /etc/keepalived/nginx_monitor.sh
# 测试一下脚本能不能执行
sh /etc/keepalived/nginx_monitor.sh
# 没报错即表示为成功
```

安装Keepalived

在两台Nginx服务上安装Keepalived 2.0.18版本

安装依赖

```
sudo yum install -y openssl-devel popt-devel libnl-devel kernel-devel gcc
```


安装Keepalived

```
cd ~
wget https://www.keepalived.org/software/keepalived-2.0.18.tar.gz
tar -xvzf keepalived-2.0.18.tar.gz
cd keepalived-2.0.18
# 安装到/usr/local/keepalived目录
./configure --prefix=/usr/local/keepalived --sysconf=/etc

sudo make & sudo make install
```

安装成功后，配置放在/etc/keepalived/目录中。

Keepalived配置

```
# - master主机
keepalived-nginx-master.conf
# - backup备机
keepalived-nginx-backup.conf
```

####

Nginx-Master配置

/etc/keepalived/keepalived-nginx-master.conf

```
# 定义一个名为monitor的脚本
vrrp_script monitor {
    # 监控脚本存放地址
    script "/etc/keepalived/nginx_monitor.sh"
    # 每隔1秒执行一次
    interval 1
}

# 定义一个vrrp示例
vrrp_instance VI_1 {
    state MASTER      #(主机为MASTER，备用机为BACKUP)
    interface enp0s3   #(HA监测网卡接口)

    virtual_router_id 61 #(主、备机的virtual_router_id必须相同，不大于255)
    priority 90        #(主、备机取不同的优先级，主机值较大，备份机值较小，值越大优先级越高)
    advert_int 1        #(VRRP Multicast广播周期秒数)

    track_script {
        monitor #(监控脚本名称)
    }
    virtual_ipaddress {
        192.168.120.200 #(VRRP HA虚拟IP)
    }
}
```

Nginx-Backup配置

/etc/keepalived/keepalived-nginx-backup.conf

```
# 定义一个名为monitor的脚本
vrrp_script monitor {
    # 监控nginx的脚本存放地址
    script "/etc/keepalived/nginx_monitor.sh"
    # 每隔1秒执行一次
    interval 1
}

# 定义一个vrrp示例
vrrp_instance VI_1 {
    state BACKUP    #(主机为MASTER，备用机为BACKUP)
    interface enp0s3  #(HA监测网卡接口)

    virtual_router_id 61 #(主、备机的virtual_router_id必须相同)
    priority 80 #(主、备机取不同的优先级，主机值较大，备份机值较小,值越大优先级越高)
    advert_int 1 #(VRRP Multicast广播周期秒数)

    track_script {
        monitor #(监控脚本名称)
    }
    virtual_ipaddress {
        192.168.120.200 #(VRRP HA虚拟IP)
    }
}
```

启动服务

启动Keepalived

```
# 启动master主机
sudo /usr/local/keepalived/sbin/keepalived -f /etc/keepalived/keepalived-nginx-master.conf
# 启动backup备机
sudo /usr/local/keepalived/sbin/keepalived -f /etc/keepalived/keepalived-nginx-backup.conf
```

查看日志

Keepalived和LVS的日志是通过syslog来输出的，通过下面的命令就能查看最新日志输出，方便我们查看错误信息

```
sudo tail -f /var/log/messages
...
nginx ##### app down
nginx ##### app down
nginx ##### app down
...
```

可以看到nginx服务还没有启动

启动两台Nginx服务

```
sudo ./nginx
```

Master-Nginx的日志输出

```
Oct  8 15:43:23 localhost Keepalived_vrrp[4127]: Script `monitor` now returning 0
Oct  8 15:43:23 localhost Keepalived_vrrp[4127]: VRRP_Script(monitor) succeeded
Oct  8 15:43:23 localhost Keepalived_vrrp[4127]: (VI_1) Entering BACKUP STATE
Oct  8 15:43:26 localhost Keepalived_vrrp[4127]: (VI_1) Entering MASTER STATE
```

注意事项

两台keepalived服务间需要通信，关闭防火墙

```
firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）
systemctl list-unit-files|grep firewalld.service
systemctl stop firewalld.service #停止firewall
systemctl disable firewalld.service #禁止firewall开机启动

[root@localhost ~]#systemctl stop firewalld.service
[root@localhost ~]#systemctl disable firewalld.service
启动一个服务：systemctl start firewalld.service
关闭一个服务：systemctl stop firewalld.service
重启一个服务：systemctl restart firewalld.service
显示一个服务的状态：systemctl status firewalld.service
在开机时启用一个服务：systemctl enable firewalld.service
在开机时禁用一个服务：systemctl disable firewalld.service
查看服务是否开机启动：systemctl is-enabled firewalld.service;echo $?
查看已启动的服务列表：systemctl list-unit-files|grep enabled
```

测试

关闭master nginx服务，服务是否仍然可用，backup服务是否替补上来。

```
ps -ef|grep nginx
root      5189      1  0 22:01 ?        00:00:00 nginx: master process
/usr/sbin/nginx -c /etc/nginx/nginx_lvs_upstream.conf
nginx     5190  5189  0 22:01 ?        00:00:00 nginx: worker process
sudo kill 5189
```

1. 关掉备机，功能完全不受影响。
2. 关掉主机，虚拟IP漂移到备机，备机开始工作。
3. 关掉主机nginx，主机监控到无nginx后，自动切换

搭建LVS高可用

只有一台LVS服务，万一挂了怎么办？需要Keepalived来做高可用，准备两台linux服务器，用来安装两台keepalived，Keepalived安装过程如下。

准备工作

做LVS高可用，我们需要准备四台服务器

服务环境设定

共需要四台服务器，两台LVS，两台RealServer，**端口号必须保持一致，设为80**，所以需要4台服务器。

设定IP环境如下

| 服务名 | IP | 端口 | 作用 |
|---------------------------------|--|----|--|
| LVS-Master Keepalived-Master | VIP 192.168.120.200 RIP 192.168.120.127 | 80 | 主LVS，对外提供虚拟IP访问 |
| LVS-Backup Keepalived-Backup | VIP 192.168.120.200 RIP 192.168.120.128 | 80 | 备份LVS，对外提供虚拟IP访问 当主LVS不可用时，VIP漂移到备份LVS 当主LVS恢复可用时，VIP漂移回主LVS |
| RealServer-Nginx1 | 192.168.120.124 | 80 | 运行nginx及tomcat服务，作为真实服务1 |
| RealServer-Nginx2 | 192.168.120.120 | 80 | 运行nginx及tomcat服务，作为真实服务2 |

LVS-Director负责将80端口的请求均衡到Nginx1、Nginx2两台真实服务器上去，接下来我们将进行配置LVS-Director的路由方式-DR，直接路由。

准备IP地址信息打印程序balancer-1.0.0.jar，<http://hostname:port/server/ip>，打印服务器的IP端口号信息。

准备LVS

在LVS-Master、LVS-Backup两台服务器上启用LVS，具体参考前面LVS安装部分内容。确保任意的LVS能够进行负载均衡。

准备后端服务

在RealServer-Nginx1、RealServer-Nginx2两台服务上启动后端服务，具体操作参考前面的环境准备。**确保各个服务端口可用正常。**

安装Keepalived

在两台LVS服务上安装Keepalived 2.0.18版本

安装依赖

```
sudo yum install -y openssl-devel popt-devel libnl-devel kernel-devel gcc
```

安装Keepalived

```
cd ~
wget https://www.keepalived.org/software/keepalived-2.0.18.tar.gz
tar -xvzf keepalived-2.0.18.tar.gz
cd keepalived-2.0.18
# 安装到/usr/local/keepalived目录
./configure --prefix=/usr/local/keepalived --sysconf=/etc

sudo make & sudo make install
```

安装成功后，配置放在/etc/keepalived/目录中。

Keepalived配置

LVS-Master配置

文件内容，参考课件资料keepalived-lvs-master.conf

```
# 全局段，故障通知邮件配置
global_defs {
    notification_email {
        root@localhost
    }
    notification_email_from root@along.com
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id keepalived_lvs
}

# 配置虚拟路由器的实例段，VI_1是自定义的实例名称，可以有多个实例段
# VI_1是自定义的实例名称
vrrp_instance VI_1 {
    #初始状态，MASTER|BACKUP
    state MASTER
    #通告选举所用端口
    interface enp0s3
    #虚拟路由的ID号（一般不可大于255）
    virtual_router_id 51
    #优先级信息 #备节点必须更低
    priority 100
    #VRRP通告间隔，秒
    advert_int 1
    #vip
    virtual_ipaddress {
        192.168.120.200
    }
}

# 设置一个virtual server段
virtual_server 192.168.120.200 80 {
    # service polling的delay时间，即检查realserver状态的间隔时间
    # 作为测试改为0，默认6
    delay_loop 0
}
```

```

#LVS调度算法: rr|wrr|lc|wlc|lb|c|sh|dh
lb_algo rr
#LVS集群模式: NAT|DR|TUN
lb_kind DR
#会话保持时间（持久连接，秒），即以用户在600秒内被分配到同一个后端realserver
#作为测试改为0
persistence_timeout 0
#健康检查用的是TCP还是UDP
protocol TCP

# real server设置段
# 后端真实节点主机的权重等设置
real_server 192.168.120.120 80 {
    #weight 1 #给每台的权重，rr无效
    #http服务
    HTTP_GET {
        url {
            path /
        }
        #连接超时时间
        connect_timeout 3
        #重连次数
        retry 3
        #重连间隔
        delay_before_retry 3
    }
}
real_server 192.168.120.124 80 {
    #weight 2
    HTTP_GET {
        url {
            path /
        }
        connect_timeout 3
        retry 3
        delay_before_retry 3
    }
}
}

```

LVS-Backup配置

文件内容，参考课件资料keepalived-lvs-backup.conf

参考LVS-Master配置，修改vrrp_instance VI_1部分的内容，state、priority两个地方。其他的保持不变。

```

# 配置虚拟路由器的实例段，VI_1是自定义的实例名称，可以有多个实例段
# VI_1是自定义的实例名称
vrrp_instance VI_1 {
    #初始状态，MASTER|BACKUP
    state BACKUP
    #通告选举所用端口
    interface enp0s3
    #虚拟路由的ID号（一般不可大于255）
    virtual_router_id 51
    #优先级信息 #各节点必须更低

```

```
priority 90
#VRRP通告间隔, 秒
advert_int 1
#vip
virtual_ipaddress {
    192.168.120.200
}
}
```

启动Keepalived

```
# 启动master主机
sudo /usr/local/keepalived/sbin/keepalived -f /etc/keepalived/keepalived-lvs-
master.conf
# 启动backup备机
sudo /usr/local/keepalived/sbin/keepalived -f /etc/keepalived/keepalived-lvs-
backup.conf
```

测试LVS高可用

关闭LVS-Master，服务是否仍然可用？

关闭RealServer-Nginx1，整个服务是否仍然可用？