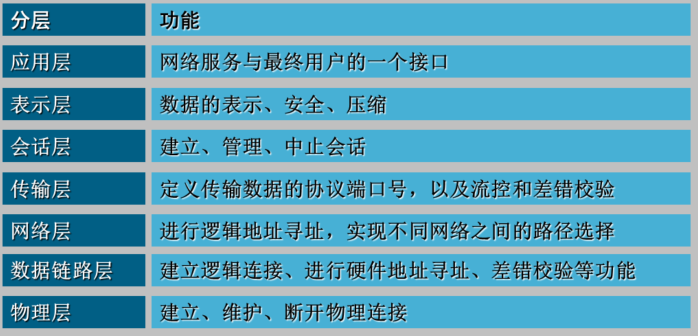
**本节所讲内容：**

* **Keepalived概述**
* **使用keepalived实现LVS-DR模式高用**

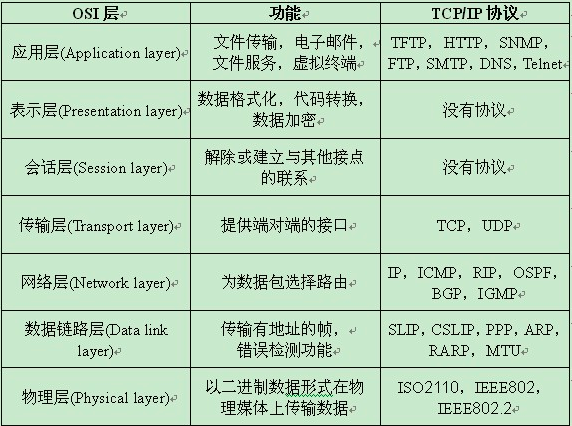
# Keepalived介绍

keepalived 是一个类似于 layer3, 4 & 5 交换机制的软件，也就是我们平时说的第 3 层、第 4 层和第 5层交换。 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。

**拓展：OSI七层**



拓展：OSI七层相关协议



# Keepalived工作原理

Layer3,4&5 工作在 IP/TCP 协议栈的 IP 层， TCP 层，及应用层,。

Layer3： Keepalived 使用 Layer3 的方式工作式时， Keepalived 会定期向服务器群中的服务器发送一个 ICMP 的数据包（既我们平时用的 Ping 程序） , 如果发现某台服务的 IP 地址没有激活，Keepalived 便报告这台服务器失效，并将它从服务器群中剔除，这种情况的典型例子是某台服务器被非法关机。 Layer3 的方式是以服务器的 IP 地址是否有效作为服务器工作正常与否的标准。

Layer4: 主要以 TCP 端口的状态来决定服务器工作正常与否。如 web server 的服务端口一般是80，如果 Keepalived 检测到 80 端口没有启动，则 Keepalived 将把这台服务器从服务器群中删除。

Layer5： Layer5 就是工作在具体的应用层了，比 Layer3,Layer4 要复杂一点，在网络上占用的带宽也要大一些。 Keepalived 将根据用户的设定检查服务器程序的运行是否正常，如果与用户的设定不相符，则 Keepalived 将把服务器从服务器群中剔除。

# Keepalived的作用与构建

1.管理 VIP VIP 会在 LVS 之间漂移

2.监控 LVS分发器

运行在主分发的 Keepalived 会以组播的形式向网络中宣告自己，即主分发器还活着，备用节点能收到。当备用节点，在一个时间单位中收不到组播，备用节点会认为主 LVS 挂了，开始接手主分发器工作，把 VIP 配给自己。

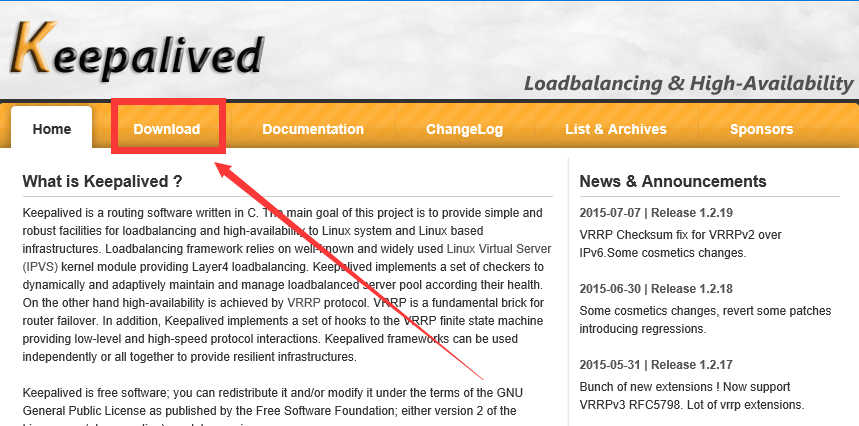
3.管理 RS

Keepalived 会每隔一个时间段去做一次类似于访问的操作如：

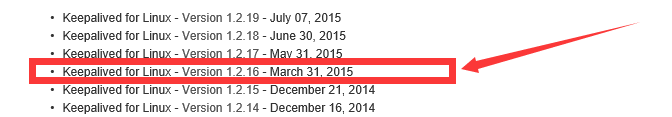
探针： elinks http://192.168.3.64 –dump

经典高可用web架构: LVS+keepalived+nginx+apache+php+eaccelerator（+nfs可选）

<http://www.keepalived.org/>



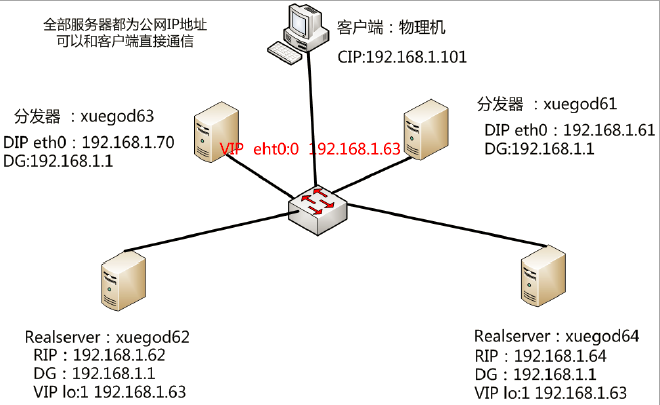
下载：<http://www.keepalived.org/download.html>



## 使用Keepalived构建LVS-DR模式的高可用集群

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **机器名称** | **机器 IP** | **网关** | **机器作用** | |
|  |  |  |  | |
| **Xuegod62.cn** | **DIP:192.168.2. 62/24** | **192.168.2.1** | **主 LVS** | |
|  | **VIP:192.168.2.111/24** |  |  |  |
|  |  |  |  | |
| **Xuegod63.cn** | **DIP:192.168.2.63/24**  **VIP:192.168.2.111/24** | **192.168.2.1** | **备 LVS** | |
|  |  |  |  |  |
| **Xuegod64.cn** | **192.168.2.64/24** | **192.168.2.1** | **RS2** | **WEB1** |
|  |  |  |  |  |
| **Xuegod65.cn** | **192.168.2.65/24** | **192.168.2.1** | **RS1** | **WEB2** |
|  |  |  |  |  |
|  |  |  |  |  |

### 网络拓扑



### 安装ipvsadm

[root@xuegod62 ~]# rpm -ivh /mnt/Packages/ipvsadm-1.27-7.el7.x86\_64.rpm

安装完成不需要做任何配置，启动方式由keepalived控制

### 安装keepalived

第一种安装方法：

Yum install –y keepalived

查看安装路径：

rpm -ql keepalived-1.3.5-1.el7.x86\_64

**第二种安装方法（源码安装）**

[root@xuegod62 ~]# tar zxvf keepalived-1.2.16.tar.gz

[root@xuegod62 ~]# cd keepalived-1.2.16

解决依赖

[root@xuegod62 ~]# yum -y install gcc openssl-devel libnfnetlink-devel

[root@xuegod62 keepalived-1.2.16]# ./configure --prefix=/usr/local/keepalived

……

Keepalived configuration

------------------------

Keepalived version : 1.2.16

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

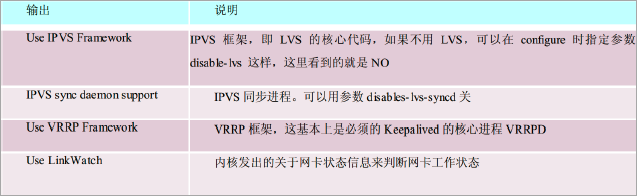
***Use VRRP Framework : Yes***

***Use VRRP VMAC : Yes***

SNMP support : No

SHA1 support : No

Use Debug flags : No

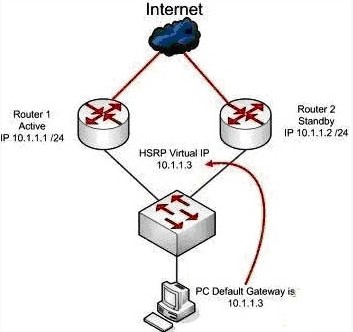
 [root@xuegod62 keepalived-1.2.16]# make && make install

拓展：VRRP/HSRP

虚拟路由冗余协议(Virtual Router Redundancy Protocol，简称VRRP)是由IETF提出的解决局域网中配置静态网关出现单点失效现象的路由协议。使用组播方式通信。

VRRP是一种路由容错协议，也可以叫做备份路由协议。一个局域网络内的所有主机都设置缺省路由（默认网关），当网内主机发出的目的地址不在本网段时，报文将被通过缺省路由发往外部路由器，从而实现了主机与外部网络的通信。当缺省路由器down掉（即端口关闭）之后，内部主机将无法与外部通信，如果路由器设置了VRRP时，那么这时，虚拟路由将启用备份路由器，从而实现全网通信。

工作流程如图：



HSRP：热备份路由器协议（HSRP）的设计目标是支持特定情况下 IP 流量失败转移不会引起混乱、并允许主机使用单路由器，以及即使在实际第一跳路由器使用失败的情形下仍能维护路由器间的连通性。换句话说，当源主机不能动态知道第一跳路由器的 IP 地址时，HSRP 协议能够保护第一跳路由器不出故障，是CISCO的私有协议！该协议中含有多台路由器，对应一个HSRP组。该组中只有一个路由器承担转发用户流量的职责，这就是活动路由器。当活动路由器失效后，备份路由器将承担该职责，成为新的活动路由器。这就是热备份的原理。

HSRP和VRRP的区别：HSRP是cisco的专有协议.在Cisco的HSRP之后,internet工程任务小组(internet engineering task force,IETF)也制定一种路由冗余协议:虚拟路由冗余协议(Virtual Router Redundancy Protocol,VRRP),目前包括Csico在内的主流厂商均在其产品中支持VRRP协议！VRRP和HSRP也有很多不同。VRRP和HSRP 的一个主要的区别在安全方面：它允许参与VRRP组的设备间建立认证机制 。另一个主要区别 ：VRRP中只有三种状态----初始状态（Initialize）、主状态（Master）、备份状态（Backup），而HSRP 有六种状态。其余在报文类型 、报文格式和通过TCP而非UDP发送的报文方面也有所不同

### 了解keepalived的文件

[root@xuegod62 keepalived-1.2.16]# cd /usr/local/keepalived/

[root@xuegod62 keepalived]# ll

bin

etc

sbin

share

#etc下有配置文件etc/keepalived/keepalived.conf

### 建立启动脚本

[root@xuegod62 keepalived]# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/

[root@xuegod62 keepalived]# grep "chkconfig" /usr/local/keepalived/etc/\* -R –color //不知道脚本位置，搜索chkconfig字段的文件可以找到！

/usr/local/keepalived/etc/rc.d/init.d/keepalived:# chkconfig: - 21 79

[root@xuegod62 keepalived]# chmod +x /etc/init.d/keepalived

因为我们使用非默认路径（/usr/local）安装 keepalived, 故需要修改几处路径，以保证 keepalived 能正常启劢, 需要修改的文件如下：

[root@xuegod62 keepalived]# vim /etc/init.d/keepalived

#. /etc/sysconfig/keepalived #此行注释

**.** **/**usr**/**local**/**keepalived**/**etc**/**sysconfig**/**keepalived #此两行添加

**export** PATH**=**"$PATH:/usr/local/keepalived/sbin" #将 keepavlied 主程序所在路径导入到环境变量 PATH 中

[root@xuegod62 keepalived]# vim /usr/local/keepalived/etc/sysconfig/keepalived

KEEPALIVED\_OPTIONS**=**"-D"

#改为

KEEPALIVED\_OPTIONS**=**"-D -f /usr/local/keepalived/etc/keepalived/keepalived.conf"

#修改指定 keepalived 要加载的配置文件，设置正确的服务启动参数

[root@xuegod62 keepalived]# ip addr

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: ens32: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:0c:29:89:ac:42 brd ff:ff:ff:ff:ff:ff

inet 192.168.2.62/24 brd 192.168.2.255 scope global ens32

valid\_lft forever preferred\_lft forever

inet 192.168.2.111/32 scope global ens32

valid\_lft forever preferred\_lft forever

### 配置Keepalived+LVS-DR模式

在这种模式下，虚拟 IP 在某时刻只能属于某一个节点，另一个节点作为备用节点存在。当主节点不可用时，备用节点接管虚拟 IP，提供正常服务。

配置参数： 节点 xuegod62(主节点)； 节点 xuegod63(备用节点) ； 虚拟 IP 192.168.2.111对外提供服务的 IP。 要求默认情况下由节点 xuegod62提供服务，当节点 xuegod62 不可用时，由节点 xuegod63 提供服务(即虚拟 IP 漂移至节点 xuegod63)。

**主节点XueGod62.cn配置**

[root@xuegod62 keepalived]# cp keepalived.conf keepalived.conf.bak

[root@xuegod62 ~]# vim /etc/keepalived/keepalived.conf

global\_defs **{**

notification\_email **{**

root**@**localhost #默认三个地址，修改可用地址

**}**

notification\_email\_from root**@**localhost

smtp\_server localhost

smtp\_connect\_timeout 30

router\_id xuegod62 #标识当前节点名字，两个节点的此项需要不相同。

**}**

#默认的配置文件中，使用第三方 smtp 服务器，但这在现实中几乎没有意义，发不出邮件，我们将其挃定为 localhost, 我们也可以将通知信息的发送交给本地 sendmail 服务处理。

vrrp\_instance apache **{** #定义一个实例，一个集群就是一个实例。 默认VI\_1 可以随意改

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 **{**

192**.**168**.**2**.**111 #指定虚拟 IP, 两个节点设置必须一样

**}**

**}**

#类似添加虚拟一个服务 ipvsadm -A -t 192.168.1.70:80 -s rr

virtual\_server 192**.**168**.**2**.**111 80 **{** #对虚拟IP63添加LVS相关内容

delay\_loop 6 #Keepalived 多长时间监测一次 RS

lb\_algo rr #分发算法

lb\_kind DR #DR 模式

nat\_mask 255**.**255**.**255**.**0

persistence\_timeout 50 #同一 IP 50秒内的请求都发到同个real server ，这个会影响LVS的 rr 调度算法， 同一 IP 超过 50 秒后，再次访问，才会被转发到另一台 real server 上。 persistence 持久性的意思

protocol TCP

# ipvsadm -a -t 192.168.2.111:80 -r 192.168.2.64 -g 添加的内容，类似这条命令

# ipvsadm -a -t 192.168.2.111:80 -r 192.168.2.65 –g 添加的内容，类似这条命令

real\_server 192**.**168**.**1**.**64 80 **{** #配置服务节点 1，需要指定 realserver 的真实 IP 地址和端口，IP不端口之间用空格隔开

weight 1 #配置服务节点的权值，权值大小用数字表示，数字越大，权值越高，设置权值大小可以为不同性能的服务器

TCP\_CHECK **{** #这段内容手动添加，爸以前的内容删除

connect\_timeout 3 #表示 3 秒无响应超时

nb\_get\_retry 3 #表示重试次数

delay\_before\_retry 3 #表示重试间隔

connect\_port 80 #检测端口

**}**

**}**

real\_server 192**.**168**.**1**.**65 80 **{**

weight 1

TCP\_CHECK **{**

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

**}**

**}**

**}**

#默认配置文件中还有两个 virtual\_server 模版，把剩下的都删除了，就可以。 如：

#virtual\_server 10.10.10.2 1358 { 。。。 }

#virtual\_server 10.10.10.3 1358 { 。。。 }

[root@xuegod62 keepalived]# systemctl restart keepalived

[root@xuegod62 keepalived]# systemctl enable keepalived

[root@xuegod62 keepalived]# ipvsadm -L –n

IP Virtual Server version 1**.**2**.**1 **(**size**=**4096**)**

Prot LocalAddress**:**Port Scheduler Flags

**->** RemoteAddress**:**Port Forward Weight ActiveConn InActConn

#注：没有看到 realserver，是因为两台 realserver 还没有开启 httpd 服务。

### 备用节点XueGod63.cn配置

### 安装ipvsadm

[root@xuegod63 ~]# rpm -ivh /mnt/Packages/ipvsadm-1.27-7.el7.x86\_64.rpm

安装完成不需要做任何配置，启动方式由keepalived控制

### 安装keepalived

第一种安装方法：

Yum install –y keepalived

和主节点xuegod62配置差不多相同，直接从xuegod62上scp拷贝配置文件

[root@xuegod62 ~]# scp /etc/keepalived/keepalived.conf 192.168.2.63:/etc/keepalived/

复制过来之后，还要做一定的修改:

[root@xuegod63~]# vim /etc/keepalived/keepalived.conf

注：红色为改动项值

router\_id xuegod63 #运行 keepalived 的机器标示符

**}**

vrrp\_instance apache **{**

state BACKUP #当前 LVS 状态为备用分发器

interface ens32

virtual\_router\_id 51

priority 90 #LVS 优先级,备的要比主的小

……

启动：

[root@xuegod63 keepalived]# systemctl restart keepalived

[root@xuegod63 keepalived]# systemctl enable keepalived

### 测试LVS IP飘移

[root@xuegod62 keepalived]# systemctl stop keepalived

[root@xuegod63 keepalived]# ip addr

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: ens32: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:0c:29:89:ac:42 brd ff:ff:ff:ff:ff:ff

inet 192.168.2.63/24 brd 192.168.2.255 scope global ens32

valid\_lft forever preferred\_lft forever

inet 192.168.2.111/32 scope global ens32

valid\_lft forever preferred\_lft forever

[root@xuegod63 keepalived]# ipvsadm -L –n

IP Virtual Server version 1**.**2**.**1 **(**size**=**4096**)**

Prot LocalAddress**:**Port Scheduler Flags

**->** RemoteAddress**:**Port Forward Weight ActiveConn InActConn

TCP 192**.**168**.**2.111**:**80 rr persistent 50

#看到192.168.1.63飘移过来了，就证明一切正常了

### 配置RS1

[root@xuegod64 ~]# vim /etc/init.d/lvsrsdr #写一个配置RS的脚本

#!/bin/bash

#description:start relserver

VIP**=**192**.**168**.**2**.**111

source **/**etc**/**init.d**/**functions #加载环境变量（可以加载所有的环境变量）

**case** **$1** **in**

**start)**

**echo** 'start LVS of Realserver DR'

**/**sbin**/**ifconfig lo**:**1 **$VIP** broadcast **$VIP** netmask 255**.**255**.**255**.**255 up

**/**sbin**/**route add **-**host **$VIP** dev lo**:**1

**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

**;;**

**stop)**

**/**sbin**/**ifconfig lo**:**1 down

**echo** 'Close LVS of Realserver DR'

**echo** '0' **>** **/**proc**/**sys**/**net**/**ipv4**/**conf**/**lo**/**arp\_ignore

**echo** '0' **>** **/**proc**/**sys**/**net**/**ipv4**/**conf**/**lo**/**arp\_announce

**echo** '0' **>** **/**proc**/**sys**/**net**/**ipv4**/**conf**/**all**/**arp\_ignore

**echo** '0' **>** **/**proc**/**sys**/**net**/**ipv4**/**conf**/**all**/**arp\_announce

**;;**

**\*)**

**echo** "Usage:$0 (start|stop)"

**exit** 1

**esac**

[root@xuegod64 ~]# chmod +x /etc/init.d/lvsrsdr

[root@xuegod64 ~]# /etc/init.d/lvsrsdr start

[root@xuegod64 ~]# echo "/etc/init.d/lvsrsdr start" >> /etc/rc.local

[root@xuegod64 ~]# ifconfig -a

……………………………………………………………………………………….

lo:1: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 192.168.2.111 netmask 255.255.255.255

loop txqueuelen 1 (Local Loopback)

[root@xuegod64 ~]# yum install -y httpd

[root@xuegod64~]# echo 192.168.2.64 > /var/www/html/index.html

[root@xuegod64 ~]# systemctl restart httpd

### 配置RS2

从Xuegod64上拷贝脚本

[root@xuegod64 ~]# scp /etc/init.d/lvsrsdr 192.168.2.65: /etc/init.d/

然后执行

[root@xuegod65 ~]# chmod +x /etc/init.d/lvsrsdr

[root@xuegod65 ~]# /etc/init.d/lvsrsdr start

[root@xuegod65 ~]# echo "/etc/init.d/lvsrsdr start " >> /etc/rc.local

[root@xuegod65 ~]# ifconfig -a

……………………………………………………………………………………….

lo:1: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 192.168.2.111 netmask 255.255.255.255

loop txqueuelen 1 (Local Loopback)

[root@xuegod65 ~]# yum install -y httpd

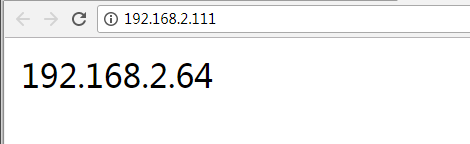
[root@xuegod65 ~]# echo 192.168.1.65 > /var/www/html/index.html

[root@xuegod65 ~]# systemctl restart httpd

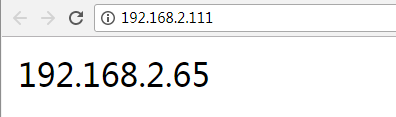
### 测试

http://192.168.2.111 #一直显示192.168.2.64的内容，等待50s之后，查看

50s之前图



50s之后图



测试主备切换，首选在主上Xueogd62上查看状态

[root@xuegod62 keepalived]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.2.111:80 rr

-> 192.168.2.64:80 Route 1 0 0

-> 192.168.2.65:80 Route 1 0 1

在备上查看Xuegod63的状态

[root@xuegod63 keepalived]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

在xuegod62上停掉keepalived, 模拟故障，在查看xuegod63

[root@xuegod62 ~]# systemctl stop keepalived

[root@xuegod63 keepalived]# ipvsadm -L –n

P Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.2.111:80 rr

-> 192.168.2.64:80 Route 1 0 0

-> 192.168.2.65:80 Route 1 0 0

[root@xuegod63 keepalived]# ip addr

……

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: ens32: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:0c:29:89:ac:42 brd ff:ff:ff:ff:ff:ff

inet 192.168.2.63/24 brd 192.168.2.255 scope global ens32

valid\_lft forever preferred\_lft forever

inet 192.168.2.111/32 scope global ens32

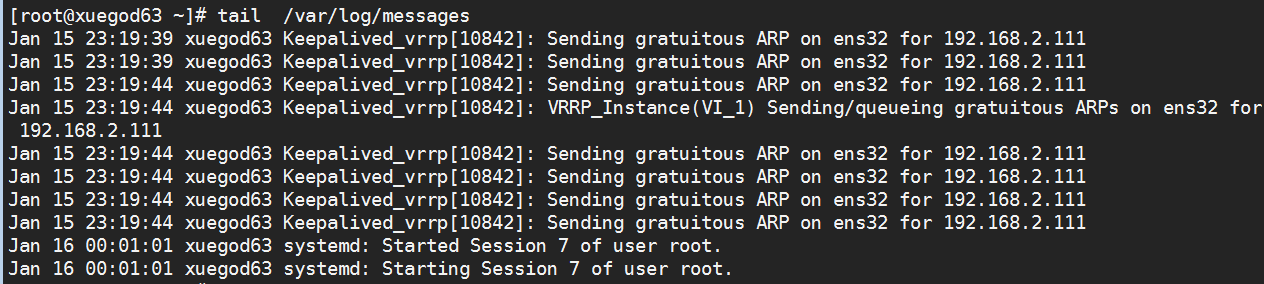
valid\_lft forever preferred\_lft forever

inet6 fe80::977:7b97:b1d1:2176/64 scope link

valid\_lft forever preferred\_lft forever

从日志看：

[root@xuegod63 keepalived]# tailf /var/log/messages



当我们重启了主上的keepalived，自动从备分发器转到主分发器上。因为备的优先级低。

[root@xuegod62 ~]# systemctl restart keepalived

测试RS容错，停掉Xueogd65的httpd之后xuegod62查看

[root@xuegod62 keepalived]# ipvsadm -L -n

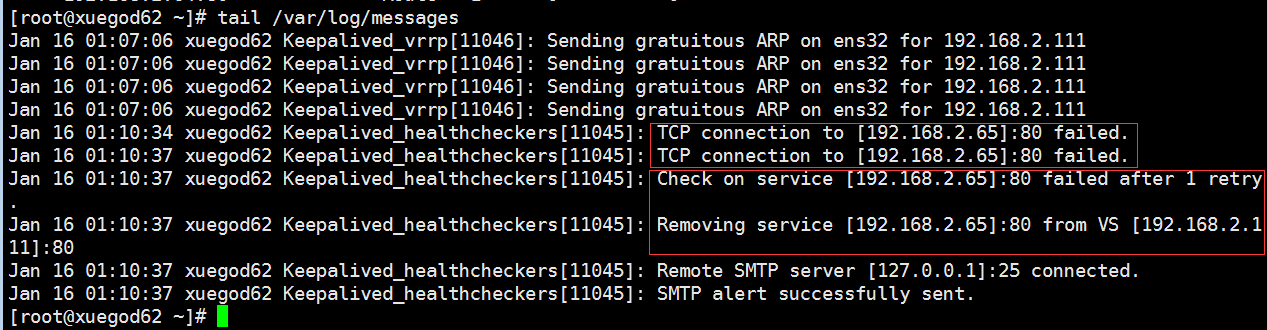
IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.2.111:80 rr

-> 192.168.2.64:80 Route 1 0 0

打开主 LVS 的日志（/var/log/message），看到滚动从 LVS 中移除192.168.2.65。 

使用 ipvsadm -Ln 可以看到刚才关闭的那台机器的 IP不在列表中了。

当 keepalived 主从优先级一样时，当主恢复后，还是要回切资源的。第一次建立主从关系时，需要10s 左右的认证时间。

附件：一个配置文件模板详解

**!** Configuration File for keepalived

global\_defs **{**

notification\_email **{**

root**@**localhost #设置报警邮件地址，即收件人地址

**}**

notification\_email\_from admin**@**localhost #设置邮件的发送地址

smtp\_server 127**.**0**.**0**.**1 #设置smtp server的ip地址

smtp\_connect\_timeout 30 #设置连接smtp server的超时时间

router\_id LTT #表示运行keepalived服务器的一个标识名字可以随便取，名字会显示在发邮件时邮件的主题信息

**}**

vrrp\_instance IN\_1 **{**

state MASTER #指定keepalived的角色，此服务为master

interface ens160 #指定监测网络的接口

virtual\_router\_id 22 #虚拟路由的标识

priority 100 #定义优先级，数字越大优先级越高，1-255之间

advert\_int 1 #设置同步检查的时间间隔，单位是秒

authentication **{** #设置验证类型

auth\_type PASS #验证类型为PASS

auth\_pass aaaa #验证的密码

**}**

virtual\_ipaddress **{**

192**.**168**.**2**.**111 #设置虚拟IP

**}**

**}**

virtual\_server 192**.**168**.**2**.**111 80 **{** #定义虚拟服务器，需指定虚拟ip和端口

delay\_loop 6 #设置健康检查时间，单位为秒

lb\_algo wlc #设置负载调度算法，此处为加权轮叫算法

lb\_kind DR #设置LVS实现负载均衡的模式

persistence\_timeout 120 #设置会话持续时间

nat\_mask 255**.**255**.**255**.**0 # 设置子网掩码

protocol TCP #设置转发协议的类型

real\_server 192**.**168**.**2**.**64 80 **{** #定义realserver，需指定ip和端口

weight 1 #指定权重

HTTP\_GET **{** #设置检测后端realserver的方式为获取http协议报文

url **{**

path **/**

status\_code 200 #状态码为200则证明后端服务器是在线的

**}**

connect\_timeout 3 #设置超时时间

nb\_get\_retry 3 #设置超时时候重试几次

delay\_before\_retry 3 #在重试的时候的时间间隔

**}**

**}**

virtual\_server 192**.**168**.**2**.**111 8090 **{** #定义虚拟服务器，需指定虚拟ip和端口

delay\_loop 6 #设置健康检查时间，单位为秒

lb\_algo wlc #设置负载调度算法，此处为加权轮叫算法

lb\_kind DR #设置LVS实现负载均衡的模式

persistence\_timeout 120 #设置会话持续时间

nat\_mask 255**.**255**.**255**.**0 #设置子网掩码

protocol TCP #设置转发协议的类型

real\_server 192**.**168**.**2**.**90 80 **{** #定义realserver，需指定ip和端口

weight 1 #指定权重

HTTP\_GET **{** #设置检测后端realserver的方式为获取http协议报文

url **{**

path **/**

status\_code 200 #状态码为200则证明后端服务器是在线的

**}**

connect\_timeout 3 #设置超时时间

nb\_get\_retry 3 #设置超时时候重试几次

delay\_before\_retry 3 #在重试的时候的时间间隔

**}**

**}**

**}**