**负载均衡方案设计及配置说明**

# 基本概念

## Nginx简介

Nginx是个轻量级、高性能的web服务。

Nginx是一个免费的，开源的，高性能的http和反响代理服务，邮件代理服务和通用的tcp/udp代理服务。

Nginx 是HTTP, SMTP, POP3和IMAP 协议的web服务和反代服务，带有高并发、高性能和低内存利用的特性。

## Keepalived简介

Keepalived 是运行在lvs 之上，它的主要功能是实现真实机的故障隔离及负载均衡器间的失败切换，提高系统的可用性

## LVS简介

LVS是一个开源的软件，可以实现LINUX平台下的简单负载均衡。LVS是Linux Virtual Server的缩写，意思是Linux虚拟服务器。目前有三种IP负载均衡技术（VS/NAT、VS/TUN和VS/DR）；八种调度算法（rr,wrr,lc,wlc,lblc,lblcr,dh,sh）。

# 负载均衡演进思路

## 第一阶段：Nginx

为保证系统功能和系统的扩展性，在系统建立之初应该有一个很好的业务拆封规则，例如我们做一个电商系统，将用户，订单，商品拆分成独立的服务。

开始阶段，系统的访问量较低时（日PV在10万左右），可以仅通过负载均衡，按照设定的访问规则，将访问不同系统的请求转发给对应的系统，并且在出现错误访问的情况下转发到错误提示页面。



## 第二阶段：Nginx + Keepalived

随着访问压力的增大(日PV将突破50万)，系统的稳定性出现问题。在单点服务还能满足业务要求的情况下，可以为负载层及其他各层引入热备，以保证在单一节点崩溃的情况下，另一个节点能自动接替其工作，为工程师解决问题赢得时间。



## 第三阶段：LVS（DR）+ Keepalived + Nginx

随着访问压力的继续增大(日PV将突破250万)，为了保证负载层足够稳定的状态下，适应访问更大的吞吐量还要应付可能的访问洪峰，我们引入LVS。

LVS负责第一层负载，然后再将访问请求转发到后端的若干台Nginx上。LVS的DR工作模式，只是将请求转到后端，后端的Nginx服务器必须有一个外网IP，在收到请求并处理完成后，Nginx将直接发送结果到请求方，不会再经LVS回发。



这里要注意的是：

有了上层的LVS的支撑Nginx就不再需要使用Keepalived作为热备方案。因为首先Nginx不再是单个节点进行负载处理，而是一个集群多台Nginx节点，LVS对于后端的服务器进行自带的基于端口的健康检查功能，即使有一两台Nginx服务器出现故障，对整个负载集群来说问题也不大。

LVS是单节点处理的，虽然LVS是非常稳定的，但是为了保证LVS更稳定的工作，我们还是需要使用Keepalived为 LVS做一个热备节点，以防不时之需。

## 第四阶段：DNS轮询 + LVS（DR）+ Keepalived + Nginx

随着系统访问量进一步增大，为了满足平均上亿的日PV访问，在对业务进行外网暴露的基础上，我们在互联网的最前端做了一个DNS轮询。然后将（对用户服务）访问压力首先分摊到两个对称LVS组上，再由每个组向下继续分拆访问压力。



和前面的方案相比，不再使用目录名分割业务系统了，而是直接将业务系统的访问使用不同的二级域名进行拆分。这样的变化有利于每个业务系统都拥有自己独立的负载均衡层。

# 整体架构

## 环境描述

共有4台服务器，安装Ubuntu 16.04 Server，分别如下：

LVS Master：10.10.8.26

LVS Backup：10.10.8.27

Nginx01：10.10.8.23

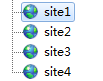
Nginx02：10.10.8.24

VIP：10.10.8.29

# 负载均衡环境搭建

## 部署后台业务服务

这里只是为了模拟业务服务，不在本文讨论之内，为演示效果，在这里，我在本机IIS中部署了4个网站：



地址：

site1： http://10.10.1.73:1212

site2： http://10.10.1.73:1213

site3： http://10.10.1.73:1214

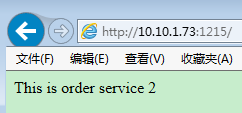
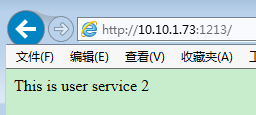
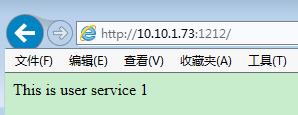
site4： http://10.10.1.73:1215

其中：

site1，site2模拟 user service 集群

site3，site4模拟 order service 集群

分别测试各站点访问正常：

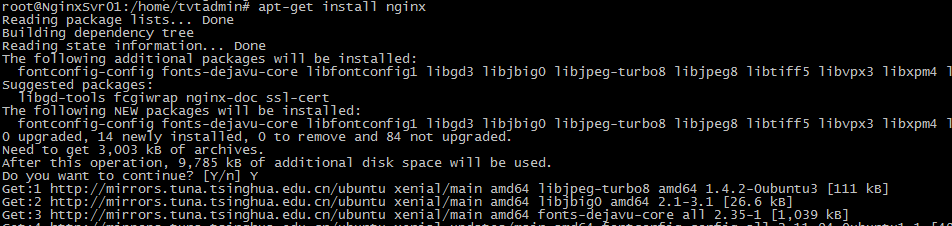


## 第一阶段架构搭建过程

### 安装Nginx

在另外两台服务器上安装Nginx：

apt-get install nginx



查看Nginx版本号：

nginx -v



apt源中nginx目前的版本号是1.10.3

确保nginx运行正常并可访问

通过浏览器访问 http://ip 验证是否可以访问Nginx默认页



### 配置Nginx

本机配置Nginx根据请求的地址，将请求分流到对应的后台业务服务集群。

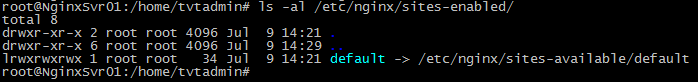
Nginx的主配置文件在 /etc/nginx/nginx.conf

可用的站点配置文件在 /etc/nginx/sites-available/ 目录

启用的站点配置文件在 /etc/nginx/sites-enabled/ 目录

我们刚才访问的网站就是当前启用的默认站点，配置文件为：

/etc/nginx/sites-enabled/default



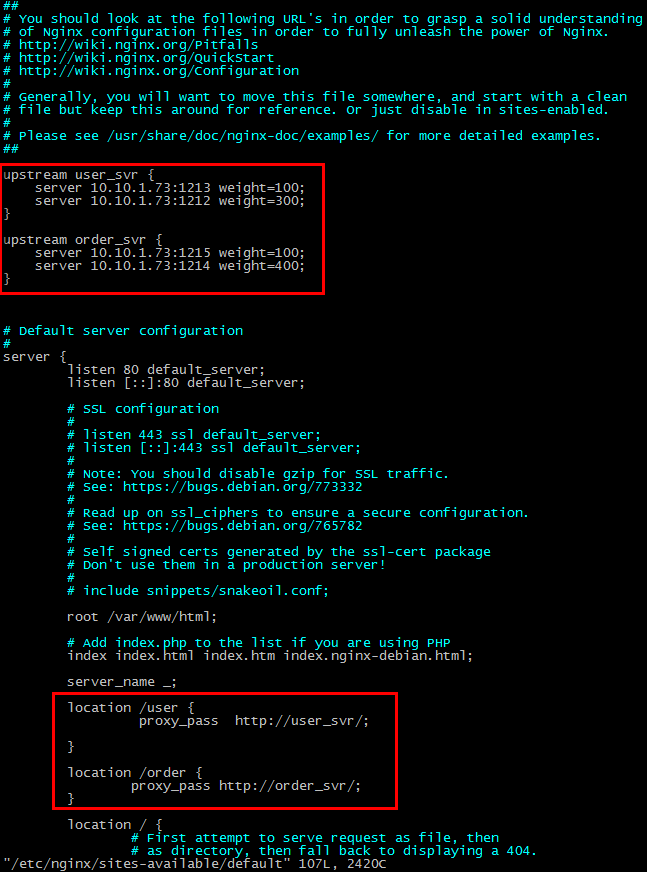
可以看到它其实是一个指向 /etc/nginx/sites-available/default 的链接文件。

先配置 Nginx01：10.10.8.23

修改/etc/nginx/sites-available/default文件：

vim /etc/nginx/sites-available/default

增加如下配置：



**location配置说明：**

语法规则： location [=|~|~\*|^~] /uri/ { … }

= 开头表示精确匹配

^~ 开头表示uri以某个常规字符串开头，理解为匹配 url路径即可。nginx不对url做编码，因此请求为/static/20%/aa，可以被规则^~ /static/ /aa匹配到（注意是空格）。

~ 开头表示区分大小写的正则匹配

~\*  开头表示不区分大小写的正则匹配

!~和!~\*分别为区分大小写不匹配及不区分大小写不匹配 的正则

/ 通用匹配，任何请求都会匹配到。

多个location配置的情况下匹配顺序为（参考资料而来，还未实际验证，试试就知道了，不必拘泥，仅供参考）：

首先匹配 =，其次匹配^~, 其次是按文件中顺序的正则匹配，最后是交给 / 通用匹配。当有匹配成功时候，停止匹配，按当前匹配规则处理请求。

**注意：**

proxy\_pass 地址末尾的/，如果不带/，转发的地址会带上location匹配地址，如：

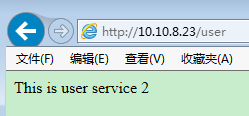
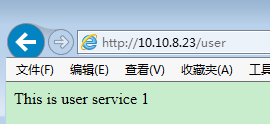
http://user\_svr/，则实际转发的是 http://10.10.1.73:1213

http://user\_svr，则实际转发的是 http://10.10.1.73:1213/user

重启Nginx服务：nginx -s reload

验证Nginx配置有效性：

访问：http://10.10.8.23/user 会将请求转发到 user service 集群



而且重复访问service 1 命中的率的确是 2 的 3 倍，和配置的权重一致

访问http://10.10.8.23/order 会将请求转发到 order service 集群，且重复访问得到的结论一致。

用相同的方式配置Nginx02：10.10.8.24

## 第二阶段架构搭建过程

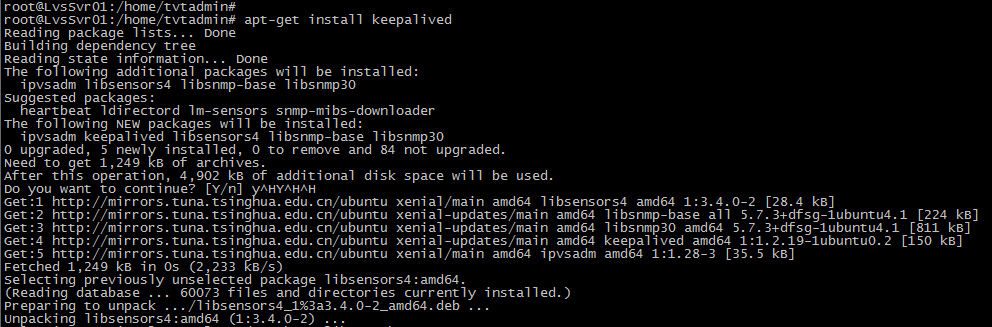
**下面以第三阶段为例，描述整个搭建过程，即：-LVS（DR）+ Keepalived + Nginx**

第一阶段，和第二阶段搭建过程已包含在此描述中，可以参考。

第四阶段涉及到域名注册和配置，后面补充。

### 安装Keepalived

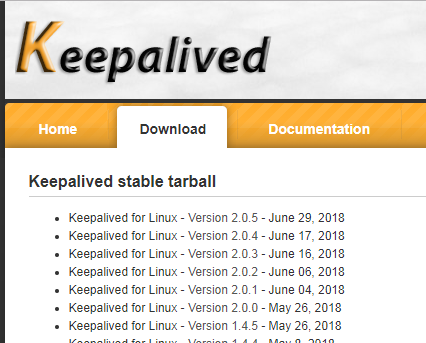
执行命令：apt-get install keepalived



Ubuntu的apt源中Keepalived的版本是1.2.19



Keepalived官网上的最新版本是 [2.0.5](http://www.keepalived.org/software/keepalived-2.0.5.tar.gz) （http://www.keepalived.org/download.html）



这里使用apt库中的版本

### 配置Keepalived（差异处红色背景标出）

安装完成后，在/etc/keepalived目录下建立keepalived.conf文件

vim /etc/keepalived/keepalived.conf

配置内容如下：

**LVS Master （10.10.8.26）**

|  |
| --- |
| global\_defs {  #notification\_email {  # acassen@firewall.loc  # failover@firewall.loc  # sysadmin@firewall.loc  #}  #notification\_email\_from Alexandre.Cassen@firewall.loc  #smtp\_server 192.168.200.1  #smtp\_connect\_timeout 30  router\_id LVS\_01  }  vrrp\_script monitor\_nginx {  script "/opt/monitor\_nginx.sh"  interval 1  Weight -20  }  vrrp\_instance VI\_1 {  state BACKUP  nopreempt  interface ens33  virtual\_router\_id 51  priority 150  advert\_int 1  mcast\_src\_ip=10.10.8.26  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  10.10.8.29  }  track\_script  {  monitor\_nginx  }  } |

**LVS Backup （10.10.8.27）**

|  |
| --- |
| global\_defs {  #notification\_email {  # acassen@firewall.loc  # failover@firewall.loc  # sysadmin@firewall.loc  #}  #notification\_email\_from Alexandre.Cassen@firewall.loc  #smtp\_server 192.168.200.1  #smtp\_connect\_timeout 30  router\_id LVS\_01  }  vrrp\_script monitor\_nginx {  script "/opt/monitor\_nginx.sh"  interval 1  Weight -20  }  vrrp\_instance VI\_1 {  state BACKUP  interface ens33  virtual\_router\_id 51  priority 100  advert\_int 1  mcast\_src\_ip=10.10.8.27  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  10.10.8.29  }  track\_script  {  monitor\_nginx  }  } |

/opt/monitor\_nginx.sh脚本

A=`ps -C nginx --no-header |wc -l`

if [ $A -eq 0 ];then

/etc/init.d/keepalived restart

echo "start nginx" >> /home/tvtadmin/check\_nginx.log

/usr/sbin/nginx

sleep 3

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

echo "stop keepalived" >> /home/tvtadmin/check\_nginx.log

killall keepalived

fi

fi

注意：

（1）同一集群的virtual\_router\_id 必须一致。

（2）使用了非抢占（nopreempt）功能，避免Master失效后，又重新上线会继续抢占master，从而发送2次切换的问题。设置时需要注意：两台state都要配置为BACKUP，其中一台priority至少比其他高50，且在priority最高的服务器配置nopreempt，其他不配置，上线后通过竞争产生Master。

### 验证Keepalived有效性

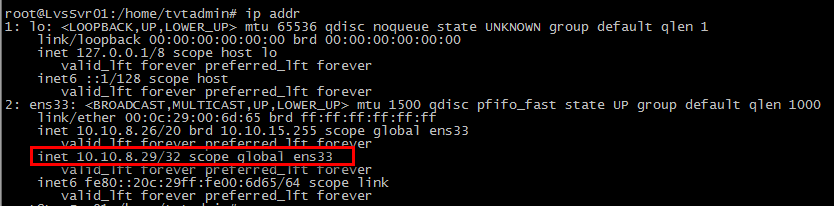
（1）在2台服务器分别启动Keepalived服务：

/etc/init.d/keepalived start

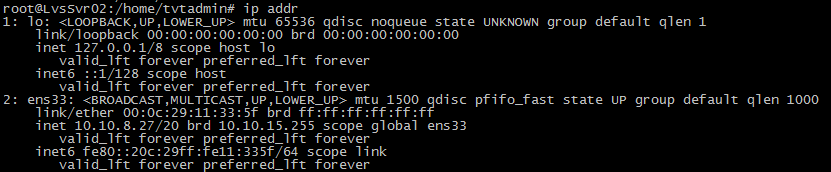


（2）分配查看两台设备的IP：

LVS Master：



LVS Backup：



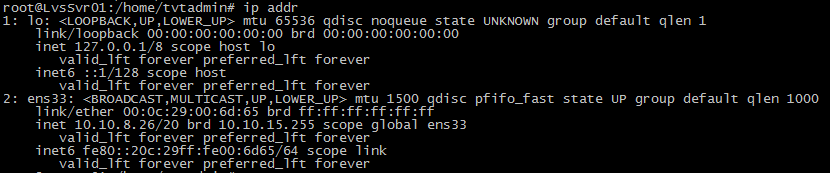
可以看到VIP（10.10.8.29）已经绑定到Master

（3）停止Master的服务，查看两台设备的IP：

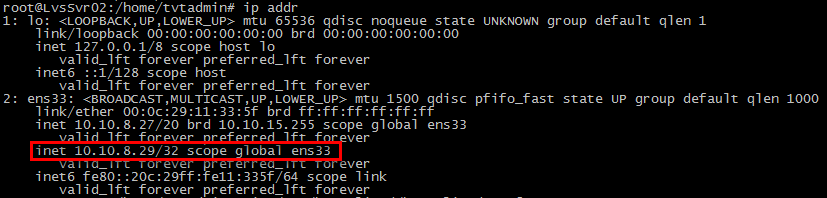
在Master上执行：

/etc/init.d/keepalived stop

LVS Master：



LVS Backup：



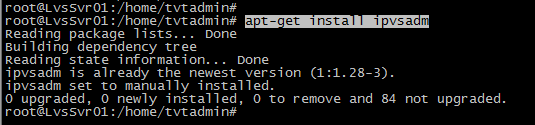
可以看到VIP（10.10.8.29）已经飘到Backup

## 第三阶段架构搭建过程

### 安装LVS

执行命令：apt-get install ipvsadm

在Ubuntu 16.04 上已默认安装了LVS



### 配置LVS

[root@lvs2 ~]# ipvsadm -C

[root@lvs2 ~]# ipvsadm -At 10.10.8.29:80 -s rr

[root@lvs2 ~]# ipvsadm -at 10.10.8.29:80 -r 10.10.8.26 -g

[root@lvs2 ~]# ipvsadm -at 10.10.8.29:80 -r 10.10.8.27 -g

[root@lvs2 ~]# ipvsadm -L –n

### 配置网卡

在两台nginx服务器上创建脚本

sudo vim /etc/init.d/realserver.sh

#!/bin/bash

# description: Config realserver lo and apply noarp

SNS\_VIP=10.10.8.29

case "$1" in

start)

ifconfig lo:0 $SNS\_VIP netmask 255.255.255.255 broadcast $SNS\_VIP

/sbin/route add -host $SNS\_VIP dev lo:0

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

sysctl -p >/dev/null 2>&1

echo "RealServer Start OK"

;;

stop)

ifconfig lo:0 down

route del $SNS\_VIP >/dev/null 2>&1

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 "RealServer Stoped"

;;

\*)

echo "Usage: $0 {start|stop}"

exit 1

esac

exit 0

通过realserver.sh start运行脚本

### 安装配置Keepalive

参照4.3在两台LVS服务器上安装Keepalive

分别配置两台Keepalive

vim /etc/keepalived/keepalived.conf

注意修改红色字体内容

global\_defs {

router\_id LVS\_DEVEL

vrrp\_skip\_check\_adv\_addr

vrrp\_strict

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

vrrp\_instance VI\_1 {

state BACKUP

nopreempt

interface enp3s0

virtual\_router\_id 50

mcast\_src\_ip 10.10.8.26

priority 100

advert\_int 1

authentication

{

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress

{

10.10.8.29

}

}

virtual\_server 10.10.8.29 80 {　　#设置虚拟服务器，需要指定虚拟ip和服务端口

delay\_loop 3　　　　#健康检查时间间隔

lb\_algo rr　　　　 #负载均衡调度算法

lb\_kind DR　　　　#负载均衡转发规则

persistence\_timeout 50　　　　#设置会话保持时间，对动态网页非常有用

protocol TCP　　　　#指定转发协议类型，有TCP和UDP两种

real\_server 10.10.8.23 80 {　　#配置服务器节点1，需要指定real server的真实IP地址和端口

weight 1　　　　#设置权重，数字越大权重越高

TCP\_CHECK { 　　　　#realserver的状态监测设置部分单位秒

connect\_timeout 3　　　　#超时时间

nb\_get\_retry 3　　　　　　#重试次数

delay\_before\_retry 3　　　　#重试间隔

connect\_port 80 　　#监测端口

}

}

real\_server 10.10.8.24 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

}

}

}

启动26,27两台服务器keepalived

/etc/init.d/keepalived start

停止23,24两台服务器keepalived

/etc/init.d/keepalived stop

浮动ip自动从23,24漂移到26,27上

# 参考

LVS原理：https://blog.csdn.net/yinwenjie/article/details/46845997

IP报文格式说明：https://blog.csdn.net/mary19920410/article/details/59035804

TCP报文格式说明：https://blog.csdn.net/mary19920410/article/details/58030147

参考：

https://www.cnblogs.com/zengkefu/p/5491922.html

http://www.austintek.com/LVS/LVS-HOWTO/HOWTO/LVS-HOWTO.LVS-DR.html

https://blog.csdn.net/column/details/load-balancing.html