

老男孩Linux云计算Keepalived高可用实践

- 2022年Keepalived集群，带你深度掌握Keepalived高可用工具。
- 该PDF文档会持续更新，有新技术会补充进去。
- 如获取干货资料，可关注公众号，如咨询\技术交流，可加群\微信，老男孩教育陪你进步。



关注公众号 领取更多干货



学习交流，加V联系

Linux技术交流QQ群784850820

附：往期干货文档笔记下载

PS：点击标题链接直接跳转

- [网络安全大厂面试真题.pdf下载](#)
- [公司网站架构图.png下载](#)
- [Linux三剑客-sed详解.pdf下载](#)
- [Git命令手册.pdf下载](#)
- [Linux速查备忘手册+知识点总结](#)

老男孩Linux云计算Keepalived高可用实践

- 1.1 什么是高可用集群？
- 1.2 高可用集群企业场景
- 1.3 高可用开源软件
 - 1.3.1 heartbeat简介
 - 1.3.2 keepalived简介
- 1.4 Keepalived介绍
- 1.5.Keepalived核心功能
- 1.6 Keepalived高可用故障切换转移原理
- 1.7 Keepalived服务搭建
 - 1)部署规划与架构图
 - 2) 部署测试节点Web01:
 - 3) 部署测试节点Web02
 - 4) 分别安装Nginx负载lb01,lb02节点
 - 5) 分别配置Nginx负载lb01,lb02节点
 - 6) 在lb01,lb02分别安装keepalived
 - 7) Keepalived配置文件说明
 - 8) 配置 lb01 keepalved.conf
 - 9) 配置 lb02 keepalved.conf
 - 10) 单实例场景与多实例
 - 11) 多实例lb01

- 12) 多实例lb02
- 1.8 Keepalived高可用裂脑
 - 1.8.1 什么是裂脑?
 - 1.8.2 裂脑的原因
 - 1.8.3 解决裂脑的常见方案
 - 1.8.4 裂脑实践
- 1.9 监听网卡上不存在IP地址问题
- 1.10 高可用服务只是针对物理服务器
- 1.11 keepalived管理lvs负载
- 1.12 keepalived视频教程
- 2.0 总结

1.1 什么是高可用集群?

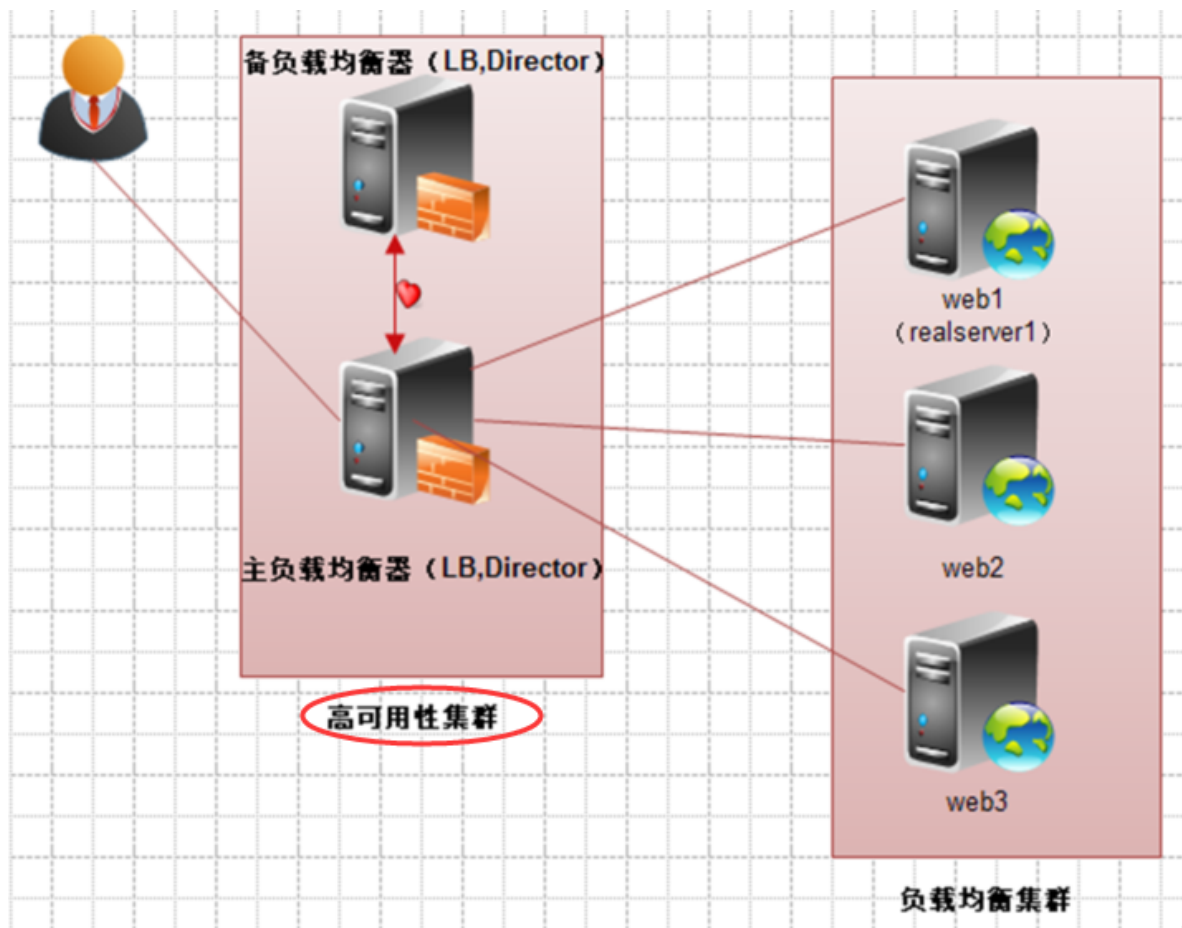
高可用性集群 (High-availability (HA) clusters) , 一般是指在集群中任意一个节点失效的情况下, 该节点上的所有任务会自动转移到其他正常的节点上。此过程并不影响整个集群的运行。

当集群中的一个节点系统发生故障时, 运行着的集群服务会迅速做出反应, 将该系统的服务分配到集群中其他正在工作的系统上运行。考虑到计算机硬件和软件的容错性, 高可用性集群的主要目的是使集群的整体服务尽可能可用。如果高可用性集群中的主节点发生了故障, 那么这段时间内将由备节点代替它。备节点通常是主节点的镜像。当它代替主节点时, 它可以完全接管主节点(包括IP地址及其他资源)的服务, 因此, 使集群系统环境对于用户来说是一致的, 即不会影响用户的访问。

高可用性集群核心作用为:

- 当一台机器宕机时, 另外一台机器接管宕机的机器的IP资源和服务资源, 提供服务。
- 常用于不易实现负载均衡的应用, 比如负载均衡器, 主数据库、主存储对之间。

高可用性集群常用的开源软件包括Keepalived、Heartbeat等, 其架构图如图所示。



1.2 高可用集群企业场景

1. 主辅负载均衡器之间，例如：LVS负载均衡器、Nginx反向代理之间。
2. 主辅存储之间、主辅数据库之间，例如：MFS,Hadoop集群主节点之间，MySQL主辅数据库之间。

1.3 高可用开源软件

1.3.1 heartbeat简介

Heartbeat项目是Linux-HA工程的一个组件，它实现了一个高可用集群系统，含有心跳服务和集群通信是高可用集群的两个关键组件，Heartbeat的核心实现简单说明如下：

- 1.以别名的方式配置VIP，命令为ifconfig，格式标签eth0:0
- 2.有3个配置文件
 - ha.cf #基本配置
 - haresources #资源配置
 - aukeys #认证
- 3.用于带有数据资源的高可用对之间，例如：数据库、存储。
- 4.heartbeat+drbd+mysql早期数据库高可用方案
 - drbd基于block复制（镜像）

1.3.2 keepalived简介

Keepalived的核心实现简单说明如下：

- 1.以辅助IP形式配置VIP，命令为ip
- 2.只有一个配置文件keepalived.conf
- 3.没有数据流动的的高可用对之间
- 4.如果有数据流动复制需求，需在keepalived之外自行解决。

1.4 Keepalived介绍

Keepalived软件起初是专为LVS负载均衡软件设计的，用来管理并监控LVS集群系统中各个服务节点的状态，后来又加入了可以实现高可用的VRRP功能。因此，Keepalived除了能够管理LVS软件外，还可以作为其他服务（例如：Nginx、Haproxy、MySQL等）的高可用解决方案软件。

1.5.Keepalived核心功能

- 1.管理LVS
- 2.对LVS节点做健康检查（基于端口和URL）
- 3.实现VRRP高可用功能

1.6 Keepalived高可用故障切换转移原理

Keepalived高可用服务对之间的故障切换转移，是通过VRRP协议（Virtual Router Redundancy Protocol 中文虚拟路由器冗余协议）来实现的。

在Keepalived服务正常工作时，主Master节点会不断地向备节点发送（多播的方式）心跳消息，用以告诉备Backup节点自己还活着，当主Master节点发生故障时，就无法发送心跳消息了，备节点也就因此无法继续检测到来自主Master节点的心跳了，进而调用自身的接管程序，接管主Master节点的IP资源及服务。而当主Master节点恢复时，备Backup节点又会释放主节点故障时自身接管的IP资源及服务，恢复到原来的备用角色。

1.7 Keepalived服务搭建

1)部署规划与架构图

1.部署规划

HOSTNAME IP 说明

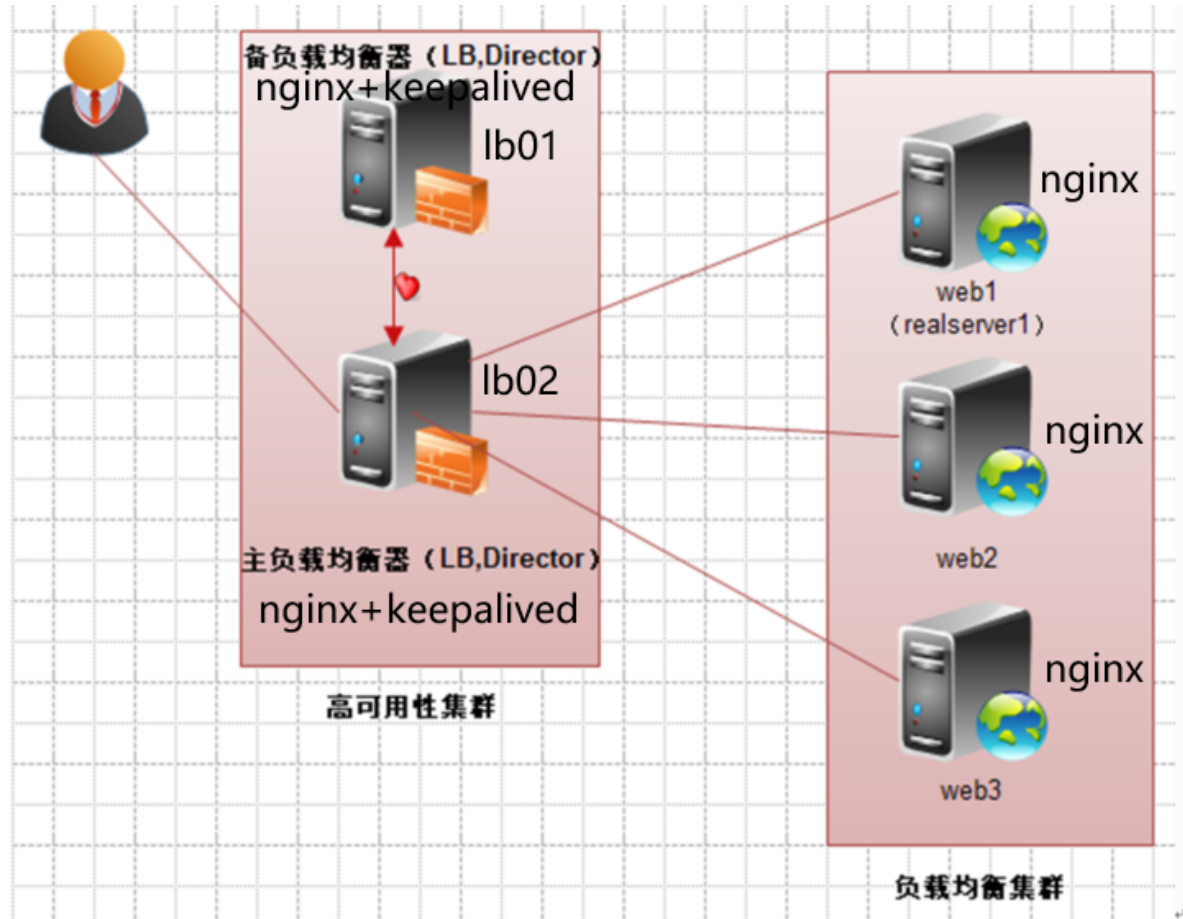
lb01 10.0.0.5 Keepalived主服务器（Nginx主负载均衡器）

lb02 10.0.0.6 Keepalived备服务器（Nginx辅负载均衡器）

web01 10.0.0.7 web01服务器（事先搭建好）

web02 10.0.0.8 web02服务器（事先搭建好）

2.部署架构图



2) 部署测试节点Web01:

```
1 [root@web01 conf.d]# cat 02_www.etiantian.org.conf
2 server {
3     listen 80;
4     server_name www.etiantian.org;
5     root /data/www;
6     location / {
7         index index.html;
8     }
9 }
10 [root@web01 conf.d]# mkdir -p /data/www
11 [root@web01 conf.d]# echo web01 >/data/www/index.html
12 [root@web01 conf.d]#
13 [root@web01 conf.d]# nginx -t
14 nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
15 nginx: configuration file /etc/nginx/nginx.conf test is successful
16 [root@web01 conf.d]# systemctl restart nginx
17 [root@web01 conf.d]# grep www /etc/hosts
18 [root@web01 conf.d]# curl -H "host:www.etiantian.org" 10.0.0.7
19 web01
```

3) 部署测试节点Web02

```
1 [root@web01 conf.d]# cat 02_www.etiantian.org.conf
2 server {
3     listen 80;
4     server_name www.etiantian.org;
```

```

5     root /data/www;
6 location / {
7     index index.html;
8 }
9 }
10 [root@web02 conf.d]# mkdir -p /data/www
11 [root@web02 conf.d]#
12 [root@web02 conf.d]# echo web01 >/data/www/index.html
13 [root@web02 conf.d]# echo web02 >/data/www/index.html
14 [root@web02 conf.d]# nginx -t
15 nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
16 nginx: configuration file /etc/nginx/nginx.conf test is successful
17 [root@web02 conf.d]# systemctl restart nginx
18 [root@web02 conf.d]# curl -H "host:www.etiantian.org" 10.0.0.8
19 web02

```

4) 分别安装Nginx负载lb01,lb02节点

```

1 1.安装nginx
2 [root@lb02 ~]# cat /etc/yum.repos.d/nginx.repo
3 [nginx-stable]
4 name=nginx stable repo
5 baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
6 gpgcheck=1
7 enabled=1
8 gpgkey=https://nginx.org/keys/nginx_signing.key
9 module_hotfixes=true
10 priority=1
11 ##
12 yum install yum-plugin-priorities -y
13 yum install nginx -y
14 systemctl start nginx
15 systemctl enable nginx

```

5) 分别配置Nginx负载lb01,lb02节点

```

1 [root@lb02 ~]# cd /etc/nginx/conf.d
2 [root@lb02 conf.d]# cat 03_www.etiantian.org.conf
3 upstream www {
4     server 172.16.1.7 weight=1 ; #默认80端口
5     server 172.16.1.8 weight=1; #默认80端口
6 }
7 server {
8     listen 80;
9     server_name www.etiantian.org;
10 location / {
11     proxy_pass http://www;
12     proxy_set_header Host $http_host;
13 }
14 }
15 nginx -t
16 systemctl restart nginx
17 #从vip 10.0.0.6/5 测试Nginx负载成功
18 [root@lb02 conf.d]# curl -H "host:www.etiantian.org" 10.0.0.6
19 web01
20 [root@lb02 conf.d]# curl -H "host:www.etiantian.org" 10.0.0.6

```

6) 在lb01,lb02分别安装keepalived

```

1 yum install keepalived -y
2 systemctl start keepalived
3 [root@lb01 ~]# ps -ef|grep keep|grep -v grep
4 root      1455      1  0 16:36 ?        00:00:00 /usr/sbin/keepalived -D
5 root      1456    1455  0 16:36 ?        00:00:00 /usr/sbin/keepalived -D
6 root      1457    1455  0 16:36 ?        00:00:00 /usr/sbin/keepalived -D
7 systemctl stop keepalived
8 rpm -qa keepalived
9 keepalived-1.3.5-19.el7.x86_64

```

7) Keepalived配置文件说明

```

1 #keepalived配置文件分三个核心：
2 GLOBAL CONFIGURATION ##全局
3 VRRPD CONFIGURATION  ##VRRP高可用
4 LVS CONFIGURATION    ##管理LVS, nginx负载用不到，此处删掉

```

```

1 [root@lb01 keepalived]# cat keepalived.conf
2 global_defs {
3     router_id lb01          #<==id为lb01，不同的keepalived.conf此ID要唯一。
4 }
5 vrrp_instance VI_1 {      #<==实例名字为VI_1，相同实例的备节点名字要和这个相同。
6     state MASTER          #<==状态为MASTER，备节点状态需要为BACKUP。
7     interface eth0        #<==通信接口为eth0，此参数备节点设置和主节点相同。
8     virtual_router_id 51  #<==实例ID为55，keepalived.conf里唯一。
9     priority 150          #<==优先级为150，备节点的优先级必须比此数字低。
10    advert_int 1          #<==通信检查间隔时间1秒。
11    authentication {
12        auth_type PASS    #<==PASS认证类型，此参数备节点设置和主节点相同。
13        auth_pass 1111    #<==密码是1111，此参数备节点设置和主节点相同。
14    }
15    virtual_ipaddress {
16        10.0.0.3/24 dev eth0 label eth0:3
17    # <==虚拟IP，即VIP，子网掩码为24位，绑定接口为eth0，别名为eth0:3，此参数备节点设置和主
18    # 节点相同。
19    }
20    # 提示：此处设置的虚拟IP为10.0.0.3，即网站域名绑定的IP。

```

8) 配置 lb01 keepalved.conf

```

1 [root@lb01 ~]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb01
4 }
5 vrrp_instance VI_1 {
6     state BACKUP
7     interface eth0
8     virtual_router_id 51
9     priority 50

```

```

10     advert_int 1
11     authentication {
12         auth_type PASS
13         auth_pass 1111
14     }
15     virtual_ipaddress {
16         10.0.0.3/24 dev eth0 label eth0:3
17     }
18 }
19 systemctl restart keepalived
20 ifconfig|egrep '10.0.0.3|10.0.0.4'

```

9) 配置 lb02 keepalived.conf

```

1 [root@lb02 ~]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb02
4 }
5
6 vrrp_instance VI_1 {
7     state MASTER
8     interface eth0
9     virtual_router_id 51
10    priority 100
11    advert_int 1
12    authentication {
13        auth_type PASS
14        auth_pass 1111
15    }
16    virtual_ipaddress {
17        10.0.0.3/24 dev eth0 label eth0:3
18    }
19 }
20 systemctl restart keepalived
21 ifconfig|egrep '10.0.0.3|10.0.0.4'
22
23 #此时，停止一端服务器。看另一端IP接管情况

```

10) 单实例场景与多实例

```

1 #单实例
2 ##一边跑所有业务，另一边处于热备干等接管状态，这种模式很浪费机器。
3 ##例如：lb01跑所有业务，lb02热备干等接管
4 ##缺点明显：浪费机器
5
6 #多实例
7 ##两边同时跑不同的服务；同时互相处于对方的热备。
8 ##例如：lb01跑www业务，lb02跑blog业务
9 ##优点：两边同时跑业务，节省资源。
10 ##缺点：需要所有流量一台落到一台机器，单台机器仍要能撑得住。

```

11) 多实例lb01

同时配置两个VIP，对应不同业务，但是同一时间跑一个，这里跑10.0.0.4这个VIP


```
1 [root@lb01 ~]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb01
4 }
5
6 vrrp_instance VI_1 {
7     state BACKUP
8     interface eth0
9     virtual_router_id 51
10    priority 50
11    advert_int 1
12    authentication {
13        auth_type PASS
14        auth_pass 1111
15    }
16    virtual_ipaddress {
17        10.0.0.3/24 dev eth0 label eth0:3
18    }
19 }
20 vrrp_instance VI_2 {
21     state MASTER
22     interface eth0
23     virtual_router_id 52
24     priority 150
25     advert_int 1
26     authentication {
27         auth_type PASS
28         auth_pass 1111
29     }
30     virtual_ipaddress {
31         10.0.0.4/24 dev eth0 label eth0:4
32     }
33 }
```

12) 多实例lb02

同时配置两个VIP，对应不同业务，但是同一时间跑一个，这里跑10.0.0.3这个VIP

```
1 [root@lb02 ~]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb02
4 }
5
6 vrrp_instance VI_1 {
7     state MASTER
8     interface eth0
9     virtual_router_id 51
10    priority 100
11    advert_int 1
12    authentication {
13        auth_type PASS
14        auth_pass 1111
15    }
16    virtual_ipaddress {
17        10.0.0.3/24 dev eth0 label eth0:3
18    }
19 }
```

```

20 vrrp_instance VI_2 {
21     state BACKUP
22     interface eth0
23     virtual_router_id 52
24     priority 50
25     advert_int 1
26     authentication {
27         auth_type PASS
28         auth_pass 1111
29     }
30     virtual_ipaddress {
31         10.0.0.4/24 dev eth0 label eth0:4
32     }
33 }

```

多实例部分的测试略。

1.8 Keepalived高可用裂脑

1.8.1 什么是裂脑？

由于某些原因，导致两台高可用服务器对在指定时间内，无法检测到对方的心跳消息，各自取得资源及服务的所有权，而此时的两台高可用服务器对都还活着并在正常运行，这样就会导致同一个IP或服务在两端同时存在而发生冲突，最严重的是两台主机占用同一个VIP地址，当用户写入数据时可能会分别写入到两端，这可能会导致服务器两端的数据不一致或造成数据丢失，这种情况就被称为裂脑。

1.8.2 裂脑的原因

- 1) 高可用服务器对之间心跳线链路故障，导致无法正常通信。
- 2) 高可用服务器对上开启了iptables防火墙阻挡了心跳消息传输。
- 3) 高可用服务器对上心跳网卡地址等信息配置不正确，导致发送心跳失败。
- 4) keepalived配置问题。

1.8.3 解决裂脑的常见方案

- 1) 同时使用串行电缆和以太网电缆连接，同时用两条心跳线路
 - 2) 当检测到裂脑时强行关闭一个心跳节点（这个功能需特殊设备支持，如Stonith、fence）。相当于备节点接收不到心跳消息，发送关机命令通过单独的线路关闭主节点的电源。
- fence和Stonith设备其实就是一个智能电源。

1.8.4 裂脑实践

见结尾视频

1.9 监听网卡上不存在IP地址问题

如果配置使用“listen 10.0.0.3:80;”的方式指定IP监听服务，而本地的网卡上没有10.0.0.3这个IP，Nginx就会报错：

```
[root@lb01 server]# nginx
```

```
nginx: [emerg] bind() to 10.0.0.3:80 failed (99: Cannot assign requested address)
```

如果要实施双主即主备同时跑不同的服务，配置文件里指定了IP监听，加入如下内核参数配置：

```
net.ipv4.ip_nonlocal_bind = 1
```

此项表示启动nginx而忽略配置中监听的IP是否存在，它同样适合Haproxy。

1.10 高可用服务只是针对物理服务器

高可用服务自身往往只能解决针对物理服务器宕机切换，但是如果服务器不宕机，而nginx负载服务宕了。此时VIP是不漂移的，因此，高可用切换就会出现問題，那如何解决呢？

功能：keepalived判断nginx负载服务是否异常，如果nginx服务异常，则关闭Nginx服务，实现服务器的切换。

1) 分别在lb01,lb02上配置如下脚本

```
1 [root@lb01 scripts]# cat chk_nginx_proxy.sh
2 #!/bin/sh
3 # 如果nginx服务停止，则关闭keep.
4 if [ `netstat -lntup|grep nginx|wc -l` -ne 1 ];then
5     systemctl stop keepalived
6 fi
7 [root@lb01 scripts]# chmod +x chk_nginx_proxy.sh
8 [root@lb01 scripts]# /server/scripts/chk_nginx_proxy.sh ##要可以执行
```

测试杀掉nginx后，会不会停止keep服务：

```
1 [root@lb01 scripts]# pkill nginx
2 [root@lb01 scripts]# sh chk_nginx_proxy.sh
3 [root@lb01 scripts]# ps -ef|grep "nginx|keep"
4 root      1769   1197  0 18:35 pts/0    00:00:00 grep -E --color=auto
   nginx|keep
```

2) 分别在lb01,lb02上配置keepalived.conf
lb01上配置keepalived.conf

```
1 [root@lb01 scripts]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb01
4 }
5 vrrp_script chk_nginx_proxy {#<==定义vrrp脚本，检测HTTP端口。
6     script "/server/scripts/chk_nginx_proxy.sh"#<==执行脚本，当nginx服务有问题，
   就停掉keepalived服务。
7     interval 2 #<==间隔2秒。
8     weight 2
9 }
10 vrrp_instance VI_1 {
11     state BACKUP
12     interface eth0
13     virtual_router_id 53
14     priority 50
15     advert_int 1
16     authentication {
17         auth_type PASS
18         auth_pass 1111
19     }
20     virtual_ipaddress {
21         10.0.0.3/24 dev eth0 label eth0:3
22     }
23     track_script {
24         chk_nginx_proxy    #<==触发检查。
25     }
```

```

26 }
27 vrrp_instance VI_2 {
28     state MASTER
29     interface eth0
30     virtual_router_id 52
31     priority 150
32     advert_int 1
33     authentication {
34         auth_type PASS
35         auth_pass 1111
36     }
37     virtual_ipaddress {
38         10.0.0.4/24 dev eth0 label eth0:4
39     }
40 }

```

lb02上配置keepalived.conf

```

1 [root@lb02 scripts]# cat /etc/keepalived/keepalived.conf
2 global_defs {
3     router_id lb02
4 }
5 vrrp_script chk_nginx_proxy {#<==定义vrrp脚本，检测HTTP端口。
6     script "/server/scripts/chk_nginx_proxy.sh"#<==执行脚本，当nginx服务有问题，
    就停掉keepalived服务。
7     interval 2 #<==间隔2秒。
8     weight 2
9 }
10
11 vrrp_instance VI_1 {
12     state MASTER
13     interface eth0
14     virtual_router_id 53
15     priority 100
16     advert_int 1
17     authentication {
18         auth_type PASS
19         auth_pass 1111
20     }
21     virtual_ipaddress {
22         10.0.0.3/24 dev eth0 label eth0:3
23     }
24     track_script {
25         chk_nginx_proxy    #<==触发检查。
26     }
27 }
28 vrrp_instance VI_2 {
29     state BACKUP
30     interface eth0
31     virtual_router_id 52
32     priority 50
33     advert_int 1
34     authentication {
35         auth_type PASS
36         auth_pass 1111
37     }
38     virtual_ipaddress {

```

```
39 | 10.0.0.4/24 dev eth0 label eth0:4
40 | }
41 | }
```

测试过程笔记略（见视频讲解）

1.11 keepalived管理lvs负载

此部分见本站的LVS教程。

1.12 keepalived视频教程

如果视频不能免费观看，可加群入群领取免费视频。

2022Keepalived高可用集群企业级实践（老男孩创始人亲授）

[点击进入学习](#)

](https://ke.oldboyedu.com/detail/term_61e549b853e90_W6Y0l7/25?fromH5=true)



Linux

高可用Keepalived 技术精讲

— 企业级案例 —

思想有多远，行动就多远



视频目录结构：

Keepalived高可用集群企业级实践（老男孩创始人亲授）-mp4

oldboy-01-集群动静分离画图复习讲解.mp4

oldboy-02-高可用集群的企业应用场景.mp4

oldboy-03-什么时候使用高可用以及开源软件对比.mp4

oldboy-04-高可用软件keepalived工作原理.mp4

oldboy-05-keepalived高可用服务环境搭建.mp4

oldboy-06-Keepalived配置文件说明.mp4

oldboy-07-测试搭建的keepalived高可用环境.mp4

oldboy-08-高可用服务对裂脑问题原因及解决方案.mp4

oldboy-09-解决nginx服务监听IP不存在的问题.mp4

2.0 总结

由于文档整理较费时费力，难免会有疏漏和不妥当地方，敬请谅解。

本文档已经发布到www.yunweibase.com/www.oldboyedu.com网站，里面还会更新更多的教程，敬请期待。

如需获取干货资料，可关注公众号获取，如咨询技术交流，可加群以及加微信，老男孩教育陪你进步！



关注公众号 领取更多干货



学习交流，加V联系

Linux技术交流QQ群784850820

****附： 往期干货文档笔记下载****

PS：点击标题链接直接跳转

- [网络安全大厂面试真题.pdf下载](#)
- [公司网站架构图.png下载](#)
- [Linux三剑客-sed详解.pdf下载](#)
- [Git命令手册.pdf下载](#)
- [Linux速查备忘手册+知识点总结](#)

成功绝非偶然，而是正确选择和持续努力后的必然！——老男孩老师 2022