LVS+keepalived+nginx+tomcat搭建高可用双机主从热备集群

作者: 王正翔

建档日期: 2020-06-07

上次更新:

版本:  1.0

2020年06月

上海汉得信息技术股份有限公司



## 文档控制

### 记录更改

| 日期 | 作者 | 版本 | 文档状态 | 更改参考 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 2020-06-07 | wzx | 1.0 | Draft | No previous version |
| 2020-09-26 | wzx | 1.1 | Updated |  |
|  |  |  |  |  |
|  |  |  |  |  |

### 审阅人

| 姓名 | 职位 |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### 分发

| 拷贝号 | 姓名 | 职位 |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**文档目录**

[文档控制 1](#_Toc53406644)

[记录更改 1](#_Toc53406645)

[审阅人 2](#_Toc53406646)

[分发 2](#_Toc53406647)

[1 系统环境准备 4](#_Toc53406648)

[1.1 虚拟机的准备 4](#_Toc53406649)

[1.2 检查所需软件包 4](#_Toc53406650)

[1.3 配置系统内核参数 5](#_Toc53406651)

[1.4 配置系统安全设置 6](#_Toc53406652)

[1.5 创建web组和web用户 7](#_Toc53406653)

[1.6 创建安装目录 7](#_Toc53406654)

[1.7 修改/etc/hosts 8](#_Toc53406655)

[1.8 安装包准备 8](#_Toc53406656)

[1.8.1 上传安装包 8](#_Toc53406657)

[2 克隆虚拟机 9](#_Toc53406658)

[2.1 保存快照 9](#_Toc53406659)

[2.2 克隆虚拟机 10](#_Toc53406660)

[2.3 调整参数 12](#_Toc53406661)

[3 安装web端服务器 15](#_Toc53406662)

[3.1 卸载自带openjdk 15](#_Toc53406663)

[3.2 安装jdk 15](#_Toc53406664)

[3.3 安装tomcat 16](#_Toc53406665)

[3.4 部署web 应用 17](#_Toc53406666)

[4 安装主从服务器 17](#_Toc53406667)

[4.1 安装nginx 17](#_Toc53406668)

[4.1.1 nginx反向代理设置 18](#_Toc53406669)

[4.1.2 nginx启停 20](#_Toc53406670)

[4.1.3 查看负载效果 21](#_Toc53406671)

[4.2 安装keepalived 21](#_Toc53406672)

[4.2.1 keepalived设置成系统服务 22](#_Toc53406673)

[4.2.2 设置LVS 23](#_Toc53406674)

[5 验证 26](#_Toc53406675)

[5.1 故障漂移 26](#_Toc53406676)

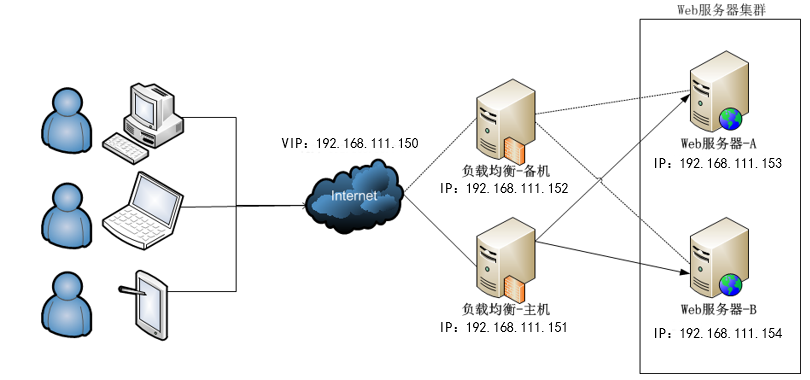
[5.2 心跳检测 30](#_Toc53406677)

[6 思考 31](#_Toc53406678)

[7 参考文档 31](#_Toc53406679)

# 系统环境准备

## 网络拓扑图



操作系统要求

|  |  |
| --- | --- |
| 系统要求 | 说明 |
| 内存 | 建议12G以上的物理内存 |
| CPU | 建议i5及以上 |
| 硬盘 | 500G及以上 |

## 虚拟机的准备

自行安装虚拟机软件 Vware workstation11

并安装Linux操作系统

镜像下载地址：http://mirrors.aliyun.com/centos/7/isos/x86\_64/CentOS-7-x86\_64-DVD-2003.iso

## 检查所需软件包

登录 root 用户，检查。如果缺少软件包，将会导致安装失败。

|  |
| --- |
| [root@localhost ~]# rpm -q gcc gcc-c++ openssl openssl-devel popt  gcc-4.1.2-51.el5  gcc-c++-4.1.2-51.el5  openssl-0.9.8e-40.el5\_11  openssl-0.9.8e-40.el5\_11  openssl-devel-0.9.8e-40.el5\_11  openssl-devel-0.9.8e-40.el5\_11  popt-1.10.2.3-36.el5\_11  popt-1.10.2.3-36.el5\_11  package popt-devel is not installed |

如果有没安装的软件包，可以使用yum命令进行下载。

可以使用yum install 命令安装，会自动下载安装对应的依赖安装包。

|  |
| --- |
| [root@localhost ~]# yum install gcc gcc-c++ openssl openssl-devel popt |

如果系统没有注册，yum 就找不到 Package 就得用rpm 到镜像文件下 Package 目录中一个个去安装了，之前系统没有注册，这里就用rpm 安装了，注意包的依赖关系，按次序安装(可以到这里下载<http://mirrors.163.com/centos/6/os/x86_64/Packages/>但不推荐使用rpm 一个个安装费时费力。建议还是用yum)

|  |
| --- |
| [root@localhost Packages]# rpm -ivh gcc-c++-4.4.6-4.el6.x86\_64.rpm |

注意使用yum 命令访问非本地源时需要系统可以访问外网。

## 配置系统内核参数

(不需要可跳过此步骤)

root 用户执行，修改配置文件 /etc/sysctl.conf，修改以下项目，如果没有可以自己添加。

|  |  |
| --- | --- |
| [root@localhost Packages]# vi /etc/sysctl.conf | |
| kernel.shmmax = 2147483648 | # 最大共享内存段大小 (安装后默认=2147483648) |
| kernel.shmall = 2097152 | # 可以使用的共享内存的总量 (安装后默认=4294967296单位页) |
| fs.aio-max-nr = 1048576 | #同时可以拥有的异步IO请求数目 |
| fs.file-max = 6815744 | # 系统中所允许的文件句柄最大数目。 |
| kernel.shmmni = 4096 | # 整个系统共享内存段的最大数目。 |
| kernel.sem = 250 32000 100 128 | # 每个信号对象集的最大信号对象数；系统范围内最大信号对象数 |
| net.ipv4.ip\_local\_port\_range = 9000 65500 | # 应用程序可使用的IPv4端口范围。 |
| net.core.rmem\_default = 262144 | # 套接字接收缓冲区大小的缺省值 |
| net.core.rmem\_max = 4194304 | # 套接字接收缓冲区大小的最大值 |
| net.core.wmem\_default = 262144 | # 套接字发送缓冲区大小的缺省值 |
| net.core.wmem\_max = 1048586 | # 套接字发送缓冲区大小的最大值 |

输入命令: vi  /etc/sysctl.conf，**按i键或insert键进入编辑模式，将下列内容加入该文件**

kernel.shmall = 2097152   
kernel.shmmax = 2147483648   
fs.file-max = 6815744   
fs.aio-max-nr = 1048576

kernel.shmmni = 4096   
kernel.sem = 250 32000 100 128   
net.ipv4.ip\_local\_port\_range = 9000 65500   
net.core.rmem\_default = 4194304   
net.core.rmem\_max = 4194304   
net.core.wmem\_default = 262144   
net.core.wmem\_max = 1048586

**复制完成后，按【ESC】键输入:wq保存退出。**

\*:wq 保存文件并退出vi

:w 保存文件但不退出vi

:q 不保存文件，退出vi

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件

**注：内核参数并非必须修改，可以根据自己实际情况而定。小技巧： 修改保存 /etc/sysctl.conf 文件之后立刻生效命令 sysctl -p。**

|  |
| --- |
| [root@localhost Packages]# sysctl –p |

提醒下：shmmax是核心参数中最重要的参数之一，用于定义单个共享内存段的最大值，shmmax设置应足够大，能在一个共享内存段下容纳下整个的SGA，设置的过低可能会导致需要创建多个共享内存段，可能导致系统性能的下降 。我分配给这台虚拟机的内存为 2G ，设置为2G。你可以根据自己的情况来设定，这是个用了比较的数据，当共享内存段被一个进程(Process)创建，操作系统检查是否被要求的共享内存段的值大于shmmax 的值 ，如果是，那么将会抛出一个错误。

## 配置系统安全设置

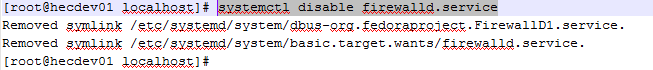
关闭防火墙服务iptables，可以直接执行如下命令：

|  |
| --- |
| [root@localhost ~]# chkconfig --list iptables  iptables 0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭  [root@localhost ~]# chkconfig iptables off |

即时启动生效

|  |
| --- |
| [root@localhost ~]# service iptables start |

Centos7可以使用命令：systemctl disable firewalld.service 来关闭防火墙



systemctl命令进程和作业管理 systemctl命令是系统服务管理器指令，它实际上将 service 和 chkconfig 这两个命令组合到一起。 <http://man.linuxde.net/systemctl>

然后，将selinux禁用即可，也就是修改/etc/selinux/config文件,检查并关闭SELinux(selinux是 Linux杰出的新安全子系统,但它的安全策略常常和软件有冲突，一般关闭它)

|  |
| --- |
| [root@localhost ~]# vi /etc/selinux/config  SELINUX=disabled |

## 创建web组和web用户

(不需要可跳过此步骤)

root 用户执行创建用户组

|  |
| --- |
| [root@localhost ~]# groupadd web |

创建用户设置密码，**这里密码设置为 web(**useradd -g web -G dba web代表将用户web 添加到用户组web 中**)**

|  |
| --- |
| [root@localhost ~]# useradd -g web -G web  [root@localhost ~]# passwd web |

## 创建安装目录

使用root用户创建目录、授权限(mkdir 创建空目录、mkdir –p 递归创建多级目录 )

\*创建一个空目录 ：mkdir test1

递归创建多个目录：mkdir -p test2/test22

创建权限为777的目录 ：mkdir -m 777 test3

将目录/u01/software及其子目录下的所有文件的所有者改为oinstall用户组下的oracle用户： chown -R oracle:oinstall /u01/software(-R或——recursive：递归处理)

chmod为更改权限命令

|  |
| --- |
| [root@localhost ~]# mkdir –p /u01/software  [root@localhost ~]# mkdir –p /u01/web  [root@localhost ~]# chmod -R 777 /u01/software  [root@localhost ~]# chmod -R 777 /u01/web |

## 修改/etc/hosts

(不需要可跳过此步骤)

|  |
| --- |
| # 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  **192.168.111.151 a.hand**  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6 |

以上ip 可以通过[root@localhost ~]$ ifconfig 查出

**a.hand 是主机名**可以通过[root@localhost ~]$ hostname 查出

|  |
| --- |
| [root@localhost Desktop]$ ifconfig  eth7 Link encap:Ethernet HWaddr 00:0C:29:FB:E1:CC  inet addr:192.168.111.151 Bcast:192.168.111.255 Mask:255.255.255.0  inet6 addr: fe80::20c:29ff:fefb:e1cc/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:147507 errors:0 dropped:0 overruns:0 frame:0  TX packets:147874 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:14568807 (13.8 MiB) TX bytes:63561560 (60.6 MiB)  lo Link encap:Local Loopback  inet addr:127.0.0.1 Mask:255.0.0.0  inet6 addr: ::1/128 Scope:Host  UP LOOPBACK RUNNING MTU:65536 Metric:1  RX packets:22334 errors:0 dropped:0 overruns:0 frame:0  TX packets:22334 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:0  RX bytes:2609554 (2.4 MiB) TX bytes:2609554 (2.4 MiB) |

## 安装包准备

### 上传安装包

安装包有 JDK：jdk-8u231-linux-x64.rpm

Nginx：nginx-1.8.0.tar.gz

Zlib：zlib-1.2.8.tar.gz

Pcre：pcre-8.37.tar.gz

Tomcat: apache-tomcat-v9.0.35.zip

Keepalived: keepalived-1.2.7.tar.gz

只需上传用FTP工具（Xftp）上传

Tomcat 上传到/u01/web 目录

其他上传到/u01/software目录下

# 克隆虚拟机

为了获得另外3台服务器，我们可以使用虚拟机克隆功能快速创建其他服务器。

在实际实施过程中可能不需要此步骤 ，另外3台服务器一般由客户IT人员提供。

## 保存快照

|  |  |
| --- | --- |
|  | 安装完毕后，建议拍摄快照方便克隆和快速恢复。 |

## 克隆虚拟机

|  |  |
| --- | --- |
|  | 点击克隆（克隆时虚拟机需要关机） |
|  | 下一步 |
|  |  |
| C:\Users\wzx\AppData\Local\Temp\1602241447(1).png |  |
|  |  |
|  |  |

## 调整参数

|  |  |
| --- | --- |
|  | **Install database softwares only** 只安装数据库软件，安装完之后创建配置数据库 |
|  | 双击eth0 进入设置 |
|  | **设置静态IP**  地址：192.168.111.151  子网掩码：255.255.255.0  网关：192.168.111.2 （和Vmnet8 中 NAT的网关保持一致） |
|  | 先禁用再激活（相当于重启网络） |
| 更改网卡eth0 配置信息  [root@localhost~]#vi /etc/sysconfig/network-scripts/ifcfg-eth0  DEVICE=eth0  BOOTPROTO=none  ONBOOT=yes  HWADDR=00:0c:29:c7:d6:40  TYPE=Ethernet  USERCTL=no  IPV6INIT=no  PEERDNS=yes  NETMASK=255.255.255.0  IPADDR=192.168.111.151  GATEWAY=192.168.111.2  重启网络  [root@localhost~]#service network restart | 上述操作也可以使用命令 完成 |

# 安装web端服务器

需要在2台虚拟主机（192.168.111.153、192.168.111.154）中安装分别安装 web服务器

## 卸载自带openjdk

（未安装openjdk 可跳过此步骤）

|  |  |
| --- | --- |
| [root@localhost~ ~]# rpm -qa |grep jdk  java-1.7.0-openjdk-headless-1.7.0.51-2.4.5.5.el7.x86\_64  java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86\_64 | 查询出系统自带的OpenJDK信息 |
| [root@localhost~ ~]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.51-2.4.5.5.el7.x86\_64  [root@localhost~ ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.51-2.4.5.5.el7.x86\_64 | 删除openJDK版本 |
| [root@localhost~ ~]# rpm -qa |grep jdk  [root@localhost~ ~]# | 再次查询确认 OpenJDK是否卸载 |

## 安装jdk

|  |  |
| --- | --- |
| [root@localhost~ ~]# cd /u01/software  [root@localhost software]# chmod +x jdk-8u231-linux-x64.rpm | 赋执行权限(已有可跳过) |
| [root@localhost software]# rpm -ivh jdk-8u231-linux-x64.rpm | 安装 |
| C:\Users\wzx\AppData\Local\Temp\1602328927(1).png | 安装完成后，会将Java安装到/usr/java下面 |
| [root@localhost~ ~]# vi /etc/profile | 配置环境变量 |
| #set java environment  export JAVA\_HOME=/usr/java/jdk1.8.0\_231-amd64  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  export PATH=$PATH:$JAVA\_HOME/bin  export JAVA\_HOME CLASSPATH PATH | 在最后加入  wq 保存并退出 |
| [root@localhost~ ~]#source /etc/profile | 使得刚添加到环境变量理解生效 |

## 安装tomcat

|  |  |
| --- | --- |
| [root@localhost~ ~]# cd /u01/web  [root@localhost web]# unzip apache-tomcat-v9.0.35.zip | 解压 |
| [root@localhost web]# chmod -R 777 /u01/web/apache-tomcat-9.0.35  [root@localhost web]# cd /u01/web/apache-tomcat-9.0.35/bin  [root@localhost bin]# ./startup.sh | 赋执行权、运行tomcat |
| C:\Users\wzx\AppData\Local\Temp\1602331729(1).png | 查看tomcat是否运行起来 |

## 部署web 应用

这里我使用的是费控系统对接汇联易的接口工程，实际实施中使用项目开发的web系统即可，本质上都是web 应用。

|  |  |
| --- | --- |
| C:\Users\wzx\AppData\Local\Temp\1602330891(1).png | 使用ftp 工具上传war 包到tomcat的webapps 目录  活动中的tomcat 会 自动解压war并运行 |
| [root@localhost~ ~]# cd /u01/web/apache-tomcat-9.0.35/webapps/hand\_hec\_hly\_service  [root@localhost hand\_hec\_hly\_service]]# vi index.jsp  <h1>IP:192.168.111.154</h1> | 为了演示效果我们在web 工程首页写明web应用所在的服务器IP  （本例在index.jsp 的<body>标签部分添加） |
| C:\Users\wzx\AppData\Local\Temp\1602332448(1).png | 查看效果 |
|  |  |

# 安装主从服务器

需要在2台虚拟主机（192.168.111.151、192.168.111.152）中安装分别安装nginx、keepalived

## 安装nginx

|  |  |
| --- | --- |
| [root@localhost~ ~]# cd /u01/software |  |
| [root@localhost software]# tar -xvf nginx-1.8.0.tar.gz  [root@localhost software]# tar -xvf pcre-8.37.tar.gz  [root@localhost software]# tar -xvf zlib-1.2.8.tar.gz | 解压nginx安装包  解压pcre包  解压zlib包 |
| [root@localhost nginx-1.8.0]#cd nginx-1.8.0  [root@localhost nginx-1.8.0]# ./configure --with-http\_stub\_status\_module --prefix=/u01/web/nginx01 --sbin-path=/u01/web/nginx01/nginx --conf-path=/u01/web/nginx01/nginx.conf --pid-path=/u01/web/nginx01/nginx.pid --error-log-path=/u01/web/nginx01/error.log --http-log-path=/u01/web/nginx01/access.log --with-pcre=/u01/software/pcre-8.37 --with-zlib=/u01/software/zlib-1.2.8 | 编译 |
| [root@localhost nginx-1.8.0]# make  [root@localhost nginx-1.8.0]# make install | 组装、安装 |

### nginx反向代理设置

反向代理（Reverse Proxy）方式是指以代理服务器来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。

|  |  |
| --- | --- |
| [root@localhost nginx-1.8.0]# vi /u01/web/nginx01/nginx.conf |  |
| #user nobody;  worker\_processes 1; #启动进程,通常设置成和cpu的数量相等  #全局错误日志及PID文件  error\_log /u01/web/nginx01/logs/error.log;  error\_log /u01/web/nginx01/logs/error.log notice;  error\_log /u01/web/nginx01/logs/error.log info;  pid /u01/web/nginx01/logs/nginx.pid;  # 工作模式及连接数上线  events {  worker\_connections 10240; #单个后台worker process进程的最大并发链接数  }  #设定http服务器,利用它的反向代理功能提供负载均衡支持  http {  include mime.types;  default\_type application/octet-stream;  sendfile on; #普通应用设为 on,如果用来进行下载等应用磁盘IO重负载应用，可设置为 off，以平衡磁盘与网络I/O处理速度，降低系统的uptime.    keepalive\_timeout 3600; #连接超时时间  proxy\_read\_timeout 3600;  client\_max\_body\_size 1000m;  gzip on; #开启gzip压缩，降低传输流量  #gzip\_vary on;  #gzip\_http\_version 1.0;  gzip\_proxied no-cache;  #gzip\_min\_length 1k;  gzip\_buffers 4 8k;  gzip\_comp\_level 9;  gzip\_types "text/plain" "text/css" "application/javascript" "image/jpeg" "image/png" "application/x-javascript" "text/javascript";  #添加tomcat列表，负载均衡的服务器都放在这。Nginx的upstream支持5种分配方式轮询、权重轮询、ip\_hash、fair、url\_hash  upstream backend{  #server tomcat地址:端口号 weight表示权值，权值越大，被分配的几率越大;  server 192.168.111.153:8080 weight=4 max\_fails=2 fail\_timeout=30s;  server 192.168.111.154:8080 weight=4 max\_fails=2 fail\_timeout=30s;  }  server {  listen 80; #监听端口  server\_name localhost; #对外提供服务的网址(域名或者ip)  location / {  root html;  index index.html index.htm; #默认访问的首页  }  location /hand\_hec\_hly\_service {  proxy\_redirect off;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_pass http://backend; #转向backend处理  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  }  #设置请求日志格式及输出路径  log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" ' '$status $body\_bytes\_sent "$http\_referer" ' '"$http\_user\_agent" $http\_x\_forwarded\_for "tomcat:" $upstream\_addr $upstream\_status $upstream\_response\_time';  access\_log /u01/web/nginx01/logs/access.log access;  } | 标黄部分自行替换 |
| [root@localhost nginx-1.8.0]# cd /u01/web/nginx01  [root@localhost nginx01]# mkdir -m 777 logs  [root@localhost nginx01]# ./nginx -t  nginx: the configuration file /u01/web/nginx01/nginx.conf syntax is ok  nginx: configuration file /u01/web/nginx01/nginx.conf test is successful | 创建nginx日志目录  检查配置 |

### nginx启停

|  |  |
| --- | --- |
| [root@localhost nginx01]# cd /u01/web/nginx01  [root@localhost nginx01]#./nginx | 启动 |
|  |  |
| [root@localhost nginx01]# ./nginx -s stop  [root@localhost nginx01]# ./nginx -s reload | 如需要 可执行停止和重载命令 |

### 查看负载效果

|  |  |
| --- | --- |
| http://192.168.111.151/hand\_hec\_hly\_service/ | 浏览器访问 |
| C:\Users\wzx\AppData\Local\Temp\1602391032(1).png | 同一地址多次刷新可访问到不同服务器上去，说明负载成功 |
|  |  |

## 安装keepalived

keepalived是基于VRRP协议实现的保证集群高可用的一个服务软件，主要功能是实现真机的故障隔离和负载均衡器间的失败切换，防止单点故障。

在了解keepalived原理之前先了解一下VRRP协议。

VRRP协议：Virtual Route Redundancy Protocol虚拟路由冗余协议。是一种容错协议，保证当主机的下一跳路由出现故障时，由另一台路由器来代替出现故障的路由器进行工作，从而保持网络通信的连续性和可靠性。

VRRP的工作工程：

(1) 虚拟路由器中的路由器根据优先级选举出 Master。 Master 路由器通过发送免费 ARP 报文，将自己的虚拟 MAC 地址通知给与它连接的设备或者主机，从而承担报文转发任务；

(2) Master 路由器周期性发送 VRRP 报文，以公布其配置信息（优先级等）和工作状况；

(3) 如果 Master 路由器出现故障，虚拟路由器中的 Backup 路由器将根据优先级重新选举新的 Master；

(4) 虚拟路由器状态切换时， Master 路由器由一台设备切换为另外一台设备，新的 Master 路由器只是简单地发送一个携带虚拟路由器的 MAC 地址和虚拟 IP地址信息的ARP 报文，这样就可以更新与它连接的主机或设备中的ARP 相关信息。网络中的主机感知不到 Master 路由器已经切换为另外一台设备。

(5) Backup 路由器的优先级高于 Master 路由器时，由 Backup 路由器的工作方式（抢占方式和非抢占方式）决定是否重新选举 Master。

VRRP优先级的取值范围为0到255（数值越大表明优先级越高）

|  |  |
| --- | --- |
| [root@localhost nginx01]# cd /u01/software  [root@localhost software]# tar -xvf keepalived-1.2.7.tar.gz | 解压 |
| [root@localhost software]# cd keepalived-1.2.7  [root@localhost software]# yum install openssl-devel  [root@localhost software]#./configure --prefix=/u01/web/keepalived  [root@localhost software]#make  [root@localhost software]#make install | 编译  openssl-devel 已安装可跳过  组装、安装 |

### keepalived设置成系统服务

|  |  |
| --- | --- |
|  |  |
| [root@localhost keepalived-1.2.7] # ln -s /u01/web/keepalived/sbin/keepalived /usr/bin/keepalived | 将keepalived命令软连接到/usr/sbin下 |
| [root@localhost keepalived-1.2.7] # cp /u01/web/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/keepalived  [root@localhost keepalived-1.2.7] #chmod 777 /etc/init.d/keepalived  [root@localhost keepalived-1.2.7] #chkconfig keepalived on | 添加启动脚本且方便用service keepalived start/stop/restart管理  添加执行权限  开机启动 |
| [root@localhost keepalived-1.2.7] # vi /etc/init.d/keepalived  # Source configuration file (we set KEEPALIVED\_OPTIONS there)  . /etc/sysconfig/keepalived  改为：  # Source configuration file (we set KEEPALIVED\_OPTIONS there)  . /u01/web/keepalived/etc/sysconfig/keepalived | 修改/etc/init.d/keepalived中的程序路径 |
| [root@localhost keepalived-1.2.7] #mkdir /etc/keepalived  [root@localhost eepalived-1.2.7] # ln -s /u01/web/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf | 默认情况下，keepalived 会读取 /etc/keepalived 下keepalived.conf 文件  如果没有建立这个文件，keepalived也不会报错，但是会发现，所创建的关于keepalived的相关参数根本就没有生效。 |
| [root@localhost keepalived-1.2.7] #service keepalived start  Starting keepalived: [ OK ]  service keepalived restart  service keepalived stop  service keepalived start  # centos 7版本用下面的命令  systemctl start keepalived  systemctl restart keepalived  systemctl stop keepalived | 启动keepalived |

### 设置LVS

LVS（Linux Virtual Server）即Linux 虚拟服务器，是一个的开源负载均衡项目，目前LVS 已经被集成到Linux 内核模块中。LVS 是四层负载均衡，也就是说建立在OSI 模型的第四层——传输层之上，传输层上有我们熟悉的TCP/UDP，LVS 支持TCP/UDP 的负载均衡。

|  |  |
| --- | --- |
| [root@localhost keepalived-1.2.7] # vi /u01/web/keepalived/etc/keepalived/keepalived.conf |  |
| global\_defs {  notification\_email {  312173702@qq.com  }  notification\_email\_from sns-lvs@gmail.com  smtp\_server 172.0.0.1  smtp\_connection\_timeout 30  router\_id LVS\_MASTER # 设置lvs的id，在一个网络应该是唯一的  }  vrrp\_script chk\_http\_port {  script "/u01/web/keepalived/check\_nginx\_pid.sh"  interval 2 #（检测脚本执行的间隔）  weight 2  }  vrrp\_instance VI\_1 {  state MASTER # 指定keepalived的角色，MASTER为主，BACKUP为备  interface eth0 # 当前进行vrrp通讯的网络接口卡(当前进行vrrp通讯的网络接口卡)  virtual\_router\_id 66 # 虚拟路由编号，主从要一致  priority 100 # 优先级，数值越大，获取处理请求的优先级越高  advert\_int 1 # 检查间隔，默认为1s(vrrp组播周期秒数)  authentication {  auth\_type PASS  auth\_pass 1111 #身份认证密码保持一致  }  track\_script {  chk\_http\_port #（调用检测脚本）  }  virtual\_ipaddress {  192.168.111.150 # 定义虚拟ip(VIP)，可多设，每行一个  }  } | 主服务器设置 |
| global\_defs {  notification\_email {  312173702@qq.com  }  notification\_email\_from sns-lvs@gmail.com  smtp\_server 172.0.0.1  smtp\_connection\_timeout 30  router\_id LVS\_DEVEL # 设置lvs的id，在一个网络应该是唯一的  }  vrrp\_script chk\_http\_port {  script "/u01/web/keepalived/check\_nginx\_pid.sh"  interval 2 #（检测脚本执行的间隔）  weight 2  }  vrrp\_instance VI\_1 {  state BACKUP # 指定keepalived的角色，MASTER为主，BACKUP为备  interface eth0 # 当前进行vrrp通讯的网络接口卡(当前进行vrrp通讯的网络接口卡)  virtual\_router\_id 66 # 虚拟路由编号，主从要一致  priority 99 # 优先级，数值越大，获取处理请求的优先级越高  advert\_int 1 # 检查间隔，默认为1s(vrrp组播周期秒数)  authentication {  auth\_type PASS  auth\_pass 1111 #身份认证密码保持一致  }  track\_script {  chk\_http\_port #（调用检测脚本）  }  virtual\_ipaddress {  192.168.111.150 # 定义虚拟ip(VIP)，可多设，每行一个  }  } | 从服务器设置 |
| [root@localhost keepalived-1.2.7] #vi /u01/web/keepalived/check\_nginx\_pid.sh | 编写check\_nginx\_pid.sh脚本 |
| #!/bin/bash  A=`ps -C nginx --no-header |wc -l`  if [ $A -eq 0 ];then  /u01/web/nginx01/nginx #重启nginx  if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then #nginx重启失败  exit 1  else  exit 0  fi  else  exit 0  fi |  |
| [root@localhost keepalived-1.2.7]#chmod 777 /u01/web/keepalived/check\_nginx\_pid.sh |  |
|  | 192.168.111.151和192.168.111.152 分别访问<http://192.168.111.150/> |
|  |  |
| [root@localhost keepalived-1.2.7] #service keepalived restart | 重启 |

# 验证

## 故障漂移

|  |  |
| --- | --- |
| [root@localhost keepalived-1.2.7] #ip addr | 查看网络信息 |
|  | 192.168.111.150 已成功绑定到192.168.111.151的eth0网卡 |
|  |  |
| [root@localhost keepalived-1.2.7] # tail -f /var/log/messages | 查看keepalived日志 |
| [root@localhost keepalived-1.2.7] #service keepalived stop | 停止192.168.111.151服务器的keepalived |
|  | 查看192.168.111.151的keepalived日志 |
| C:\Users\wzx\AppData\Local\Temp\1602428881(1).png | 查看192.168.111.152的keepalived日志 显示已进入MATER 状态 |
| [root@localhost keepalived-1.2.7] #ip addr |  |
|  |  |
|  | 192.168.111.150 已成功绑定到192.168.111.152的eth0网卡 |
| [root@localhost keepalived-1.2.7] #service keepalived start | 恢复192.168.111.151服务器的keepalived |
|  | 查看192.168.111.151的keepalived日志 显示已进入MATER 状态 |
|  | 查看192.168.111.152的keepalived日志 显示已进入BACKUP 状态 |
|  |  |
|  | 192.168.111.150 已成功绑定到192.168.111.151的eth0网卡 |
|  | <http://192.168.111.150/hand_hec_hly_service/>  始终可以正常访问 |

## 心跳检测

keepalived 在LVS基础上实现了心跳检测，如果服务器发生异常停止服务，可以尝试自动运行重启脚本，如果多次重试还是失败，将会发送邮件给运维人员。

|  |  |
| --- | --- |
| [root@localhost keepalived-1.2.7] #cd /u01/web/nginx01  [root@localhost keepalived-1.2.7] #./nginx –s stop  [root@localhost keepalived-1.2.7] #ps –ef|grep nginx | 检查lvs 检测脚本功能是否正常 （/u01/web/keepalived/check\_nginx\_pid.sh） |
|  | 从nginx 的pid 上看，可以发现nginx 是被重启了。 |

# 思考

互为主备方案：考虑到keepalived高可用，备节点的机器基本上属于空闲状态，很浪费硬件资源，所以我们可以让keepalived互为主备，跑多个实例。

Keepalived+LVS+MySQL双主复制实现读写负载均衡及高可用：既然keepalived 工作在OSI 第4层传输层

是不是也可以做数据库级的高可用。

# 参考文档

https://blog.csdn.net/gui951753/article/details/80316565

https://www.cnblogs.com/youzhibing/p/5021224.html

https://www.cnblogs.com/edisonchou/p/4281978.html

https://blog.csdn.net/bjgaocp/article/details/88417428