# Keepalived+Nginx高可用安装部署（含Nginx+Tomcat负载均衡）

## 规划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIP | IP | 主机名 | Nginx端口 | Tomcat端口 | 主从 |
| 192.168.56.101 | 192.168.56.103 | leo.dis2 | 88 | 8080 | 主 |
| 192.168.56.104 | leo.dis3 | 88 | 8080 | 从 |

CentOS6.8

Nginx 1.13.1

Keepalived 1.3.5

Tomcat7.0.78

JDK1.8.0\_131

## 安装Nginx

使用root用户，安装依赖包：

|  |
| --- |
| yum install gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel openssl openssl-devel |

把Nginx包放到/usr/local/src内，解压缩并安装：

|  |
| --- |
| tar -zxvf nginx-1.13.1.tar.gz  cd nginx-1.13.1  ./configure --prefix=/usr/local/nginx  make && make install |

## 配置Nginx

vi /usr/local/nginx/conf/nginx.conf

在开头增加

**user root;**

监听端口80改为**88**。

vi /usr/local/nginx/html/index.html

分别给两台服务器的Nginx页面增加103/104的标识。

启动Nginx

/usr/local/nginx/sbin/nginx

# 重新载入配置文件

/usr/local/nginx/sbin/nginx -s reload

# 重启 Nginx

/usr/local/nginx/sbin/nginx -s reopen

# 停止 Nginx

/usr/local/nginx/sbin/nginx -s stop

## 打开防火墙端口

vi /etc/sysconfig/iptables

|  |
| --- |
| ## Nginx  -A INPUT -m state --state NEW -m tcp -p tcp --dport 88 -j ACCEPT |

重启iptables

service iptables restart

1)、重启后永久性生效：

开启：chkconfig iptables on

关闭：chkconfig iptables off

2)、即时生效，重启后失效：

开启：service iptables start

关闭：service iptables stop

## Nginx开机启动

vi /etc/rc.local

添加 /usr/local/nginx/sbin/nginx

## Keepalived安装

使用root用户，安装依赖包：

|  |
| --- |
| yum install openssl-devel libnl3-devel ipset-devel iptables-devel libnfnetlink-devel popt popt-static popt-devel gcc kernel-headers kernel-devel net-snmp-devel -y |

把Keepalived包放到/usr/local/src内，解压缩并安装：

|  |
| --- |
| tar -zxvf keepalived-1.3.5.tar.gz  cd keepalived-1.3.5  ./configure --prefix=/usr/local/nginx  make && make install |

## 7、设置为服务并开机启动

**接下来是跟很多教程不一样的地方。**

在很多教程里，安装完毕后下一步应该是复制、链接一些文件，以便把keepalived设置成系统服务，但是请注意/usr/local/keepalived/etc/rc.d/init.d/keepalived这个文件，教程上都说有这个文件，复制到/etc/init.d/即可，可是在我安装过程中，没有这个目录和文件，所以要按照以下的方法来处理：

|  |
| --- |
| mkdir -p /etc/keepalived  cd /usr/local/keepalived/  ln -s /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/  ln -s /usr/local/keepalived/sbin/keepalived /usr/sbin/  ln -s /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/  touch /etc/rc.d/init.d/keepalived  chmod +x /etc/rc.d/init.d/keepalived  vi /etc/rc.d/init.d/keepalived |

keepalived脚本内容：

|  |
| --- |
| 1. #!/bin/sh 2. # 3. # keepalived High Availability monitor built upon LVS and VRRP 4. # 5. # chkconfig: - 86 14 6. # description: Robust keepalive facility to the Linux Virtual Server project \ 7. # with multilayer TCP/IP stack checks. 8. ### BEGIN INIT INFO 9. # Provides: keepalived 10. # Required-Start: $local\_fs $network $named $syslog 11. # Required-Stop: $local\_fs $network $named $syslog 12. # Should-Start: smtpdaemon httpd 13. # Should-Stop: smtpdaemon httpd 14. # Default-Start: 15. # Default-Stop: 0 1 2 3 4 5 6 16. # Short-Description: High Availability monitor built upon LVS and VRRP 17. # Description: Robust keepalive facility to the Linux Virtual Server 18. # project with multilayer TCP/IP stack checks. 19. ### END INIT INFO 20. # Source function library. 21. . /etc/rc.d/init.d/functions 22. exec="/usr/sbin/keepalived" 23. prog="keepalived" 24. config="/etc/keepalived/keepalived.conf" 25. [ -e /etc/sysconfig/$prog ] && . /etc/sysconfig/$prog 26. lockfile=/var/lock/subsys/keepalived 27. start() { 28. [ -x $exec ] || exit 5 29. [ -e $config ] || exit 6 30. echo -n $"Starting $prog: " 31. daemon $exec $KEEPALIVED\_OPTIONS 32. retval=$? 33. echo 34. [ $retval -eq 0 ] && touch $lockfile 35. return $retval 36. } 37. stop() { 38. echo -n $"Stopping $prog: " 39. killproc $prog 40. retval=$? 41. echo 42. [ $retval -eq 0 ] && rm -f $lockfile 43. return $retval 44. } 45. restart() { 46. stop 47. start 48. } 49. reload() { 50. echo -n $"Reloading $prog: " 51. killproc $prog -1 52. retval=$? 53. echo 54. return $retval 55. } 56. force\_reload() { 57. restart 58. } 59. rh\_status() { 60. status $prog 61. } 62. rh\_status\_q() { 63. rh\_status &>/dev/null 64. } 65. case "$1" in 66. start) 67. rh\_status\_q && exit 0 68. $1 69. ;; 70. stop) 71. rh\_status\_q || exit 0 72. $1 73. ;; 74. restart) 75. $1 76. ;; 77. reload) 78. rh\_status\_q || exit 7 79. $1 80. ;; 81. force-reload) 82. force\_reload 83. ;; 84. status) 85. rh\_status 86. ;; 87. condrestart|try-restart) 88. rh\_status\_q || exit 0 89. restart 90. ;; 91. \*) 92. echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload}" 93. exit 2 94. esac 95. exit $? |

注意脚本里面的目录，要跟上面cp、ln的目录一致。

设置服务开机启动：

|  |
| --- |
| chkconfig --add keepalived  chkconfig --level 35 keepalived on |

## 8、设置Keepalived

主节点（192.168.56.103）：

vi /etc/keepalived/keepalived.conf

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  router\_id leo.dis2 #主机名  }  vrrp\_script chk\_nginx {  script "/etc/keepalived/nginx\_check.sh"  interval 2  weight -20  }  vrrp\_instance VI\_1 {  state MASTER #主节点 建议设置为BACKUP，具体解释见后  interface eth1 #指定监测的网卡  virtual\_router\_id 51 #虚拟路由ID，节点保持一致  mcast\_src\_ip 192.168.56.103 #本机IP  priority 100 #权重  nopreempt #设置为不抢占资源  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 1357924680  }  track\_script {  chk\_nginx  }  virtual\_ipaddress {  192.168.56.101#虚拟IP  }  } |

从节点（192.168.56.104）：

vi /etc/keepalived/keepalived.conf

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  router\_id leo.dis3  }  vrrp\_script chk\_nginx {  script "/etc/keepalived/nginx\_check.sh"  interval 2  weight -20  }  vrrp\_instance VI\_1 {  state BACKUP  interface eth1  virtual\_router\_id 51  mcast\_src\_ip 192.168.56.104  priority 90  nopreempt  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 1357924680  }  track\_script {  chk\_nginx  }  virtual\_ipaddress {  192.168.56.101  }  } |

注意：**nopreempt**这个选项要解释解释。

这个设置项，允许一个priority比较低的节点作为master，即使有priority更高的节点启动。

通常如果master服务死掉后backup会变成master，但是当master服务又好了的时候 master此时会抢占VIP，这样就会发生两次切换对业务繁忙的网站来说是不好的。

所以我们要在配置文件加入 nopreempt 非抢占，但是这个参数只能用于state 为backup，故我们在用的时候最好master 和backup的state都设置成backup 让其通过priority来竞争。

**但是我在测试的时候，不管怎么配置， VIP资源总是会被抢占，后来发现需要在防火墙上设置可以组播，即：**

|  |
| --- |
| vim /etc/sysconfig/iptables  增加：  -A INPUT -d 224.0.0.18 -j ACCEPT |

**重启防火墙。**

监测Nginx脚本：

|  |
| --- |
| #!/bin/bash  A=`ps -C nginx –no-header |wc -l`  if [ $A -eq 0 ];then  /usr/local/nginx/sbin/nginx  sleep 2  if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then  killall keepalived  fi  fi |

脚本赋权：

|  |
| --- |
| chmod +x /etc/keepalived/nginx\_check.sh |

启动Keepalived：

|  |
| --- |
| service keepalived start |

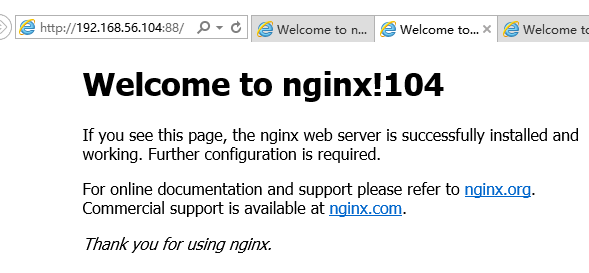
## 9、验证

启动Nginx、Keepalived。

访问<http://192.168.56.103:88/>



访问<http://192.168.56.104:88/>



访问<http://192.168.56.101:88/>



## 10、Nginx+Tomcat负载均衡

Nginx+Tomcat负载均衡的教程特别多，不再多介绍，简单提几个点。

1. 两个nginx.conf修改以下几处：

|  |
| --- |
| #增加  upstream tomcat.com{  #Tomcat1  192.168.56.103:8080 weight=1;  #Tomcat2  192.168.56.104:8080 weight=1;  }  #修改  location / {  #root html;  #index index.html index.htm;  proxy\_pass <http://tomcat.com>;  proxy\_redirect default;  } |

1. JDK要安装，防火墙要打开端口。
2. 修改两个Tomcat/webapps/ROOT/index.jsp，分别做上标识。

访问<http://192.168.56.101:88>，即可看到不同标识的Tomcat主页。