nginx高可用实现部署详细文档

**系统**

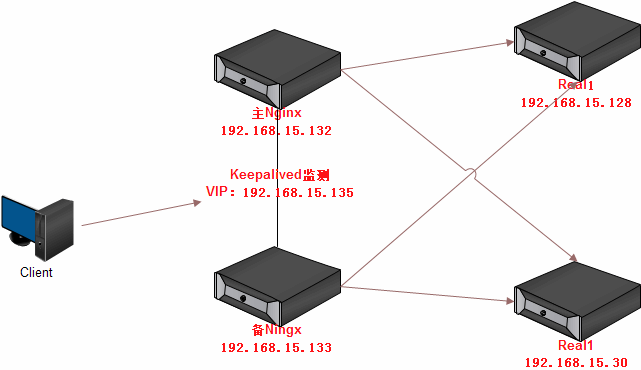
两台Nginx：

CentOS6.5 x86\_64

两台tomcat：

CentOS6.5 x86\_64

**拓扑**

[](http://static.oschina.net/uploads/space/2015/1229/110715_qE53_1437015.gif)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 机器IP | 安装软件 | 角色 | 虚拟ip | 描述 |
| 192.168.15.132 | Nginx、keepalived | Nginx主机 | 192.168.15.135 | 反向代理到tomcat1和tomcat2 |
| 192.168.15.133 | Nginx、keepalived | Nginx备机 | 主机挂了切换虚拟ip  192.168.15.135 | 反向代理到tomcat1和tomcat2 |
| 192.168.15.128 | Tomcat | Tomcat1 | 无 |  |
| 192.168.15.30 | Tomcat | Tomcat2 | 无 |  |

**IP地址**

nginx(主LB):192.168.15.132

nginx(备LB):192.168.15.133

VIP地址:192.168.15.135

Real1的IP:192.168.15.128

Real2的IP:192.168.15.30

# 一．修改启动web服务器

在192.168.15.128和192.168.15.30上分别解压修改tomcat的index页面：

vi webapps/ROOT/index.jsp

修改页面html部分

<html>

<head>

</head>

<body>

SessionID:<%=session.getId()%>

<br/>

SessionIP:<%=request.getServerName()%>

<br/>

<h1>tomcat1 page</h1>

</body>

</html>

这样可以通过访问index来判断访问是不是轮询的

# 二．安装配置nginx

## 1.安装GCC编译器等工具：

**yum install -y gcc gcc-c++ autoconf automake libtool make openssl openssl-devel** pcre pcre-devel

## 2.下载安装Nginx:

wget <http://nginx.org/download/nginx-1.6.3.tar.gz>

注：这里也可以下载tengine压缩包，比一般nginx多一些功能

tar -zxvf nginx-1.6.3.tar.gz   
cd nginx-1.6.3/

./configure --prefix=/usr/local/nginx

--sbin-path=/usr/local/nginx/sbin/nginx

--conf-path=/usr/local/nginx/conf/nginx.conf

--pid-path=/usr/local/nginx/logs/nginx.pid \

--with-http\_ssl\_module \

--with-http\_stub\_status\_module \

--with-http\_gzip\_static\_module \

make && make install

注：查询"./configure --help"相关模块，按需求指定启用

## 3.配置Nginx.conf

配置文件，二个nginx负载均衡器的文件一样

这里面配置较多，其中比较有用的已经标红了，可以直接将此部分配置在默认的nginx的配置文件里面即可。

|  |
| --- |
| user nobody;  worker\_processes  1;  error\_log  /usr/local/nginx/logs/error.log notice;  pid        /usr/local/nginx/logs/nginx.pid;  worker\_rlimit\_nofile 51200;  events {      use epoll;      worker\_connections  51200;  }  http {      include       mime.types;      default\_type  application/octet-stream;      log\_format  main  '$remote\_addr - $remote\_user [$time\_local] "$request" '                        '$status $body\_bytes\_sent "$http\_referer" '                        '"$http\_user\_agent" "$http\_x\_forwarded\_for"';      access\_log  logs/access.log  main;      server\_names\_hash\_bucket\_size 128;      client\_header\_buffer\_size 32k;      large\_client\_header\_buffers 4 32k;      client\_max\_body\_size 8m;      sendfile        on;      tcp\_nopush     on;      server\_tokens off;      keepalive\_timeout  60;      fastcgi\_connect\_timeout 300;      fastcgi\_send\_timeout 300;      fastcgi\_read\_timeout 300;      fastcgi\_buffer\_size 64k;      fastcgi\_buffers 4 64k;      fastcgi\_busy\_buffers\_size 128k;      fastcgi\_temp\_file\_write\_size 128k;      gzip  on;  **upstream backend**  **{**  **server 192.168.15.128;**  **server 192.168.15.130;**  **}**      server {          listen       80;          server\_name  192.168.15.135;          location / {              root   html;              index  index.php index.html index.htm;              proxy\_redirect off;              proxy\_set\_header Host $host;              proxy\_set\_header X-Real-IP $remote\_addr;              #后端的Web服务器可以通过X-Forwarded-For获取用户真实IP              proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  **proxy\_pass http://backend;**          }          error\_page   500 502 503 504  /50x.html;          location = /50x.html {              root   html;          }            location /nginx\_status {              stub\_status on;              auth\_basic "NginxStatus";              auth\_basic\_user\_file /usr/local/nginx/htpasswd;              #allow 127.0.0.1;              #deny all;          }          location ~\* \.(ini|docx|txt|doc|pdf)$ {          #禁止访问文档性文件          root /usr/share/nginx/html;          deny all;          }          location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf|js|html|htm|css)$ {          root /home/image;          proxy\_store on;          proxy\_store\_access user:rw group:rw all:rw;          proxy\_temp\_path /home/image;          if ( !-e $request\_filename) {             proxy\_pass  http://backend;          }      }  }  } |

# 三．安装及配置keepalived

## 1.安装keepalived

在两台nginx服务器张安装keepalived：

wget http://www.keepalived.org/software/keepalived-1.2.15.tar.gz

tar -zxvf keepalived-1.2.15.tar.gz

cd keepalived-1.2.15

**./configure --sysconf=/etc/  --with-kernel-dir=/usr/src/kernels/2.6.32-573.8.1.el6.x86\_64**

**make && make install**

ln -s /usr/local/sbin/keepalived  /sbin/

#这一步很重要，不执行ln -s会报错“Starting keepalived: /bin/bash: keepalived: command not found”

service keepalived start

**上述步骤也可以直接通过yum –y install keepalived 来代替，不同安装方式而已，不是重点**

二台Nginx上keepalived.conf配置文件如下，配置完成后分别service keepalived start启动｡检验keepalived配置是否成功

**两台keepalived的配置如下：可以看出两者之间的区别仅仅是主备的权重不同，主为100备为66，其余一些有作用的配置已经用红色标注**

**主：**

|  |
| --- |
| global\_defs {     notification\_email {       test@163.com     }     notification\_email\_from keepalived@localhost     smtp\_server 127.0.0.1     smtp\_connect\_timeout 30     router\_id LVS\_MASTER  }  vrrp\_script **chk\_http\_port** {  script "/usr/local/src/check\_nginx\_pid.sh"  interval 2          #（检测脚本执行的间隔）  weight 2  }  vrrp\_instance VI\_1 {     #state MASTER      state BACKUP      nopreempt  **#设置非抢占模式时，修改“state MASTER”为“state BACKUP”，添加“nopreempt“**      interface bond0      virtual\_router\_id 51      priority 100      advert\_int 1      authentication {          auth\_type PASS          auth\_pass 1111  }  track\_script {  **chk\_http\_port**            #（调用检测脚本）  }   virtual\_ipaddress {       192.168.15.135   }  } |

**备：**

|  |
| --- |
| global\_defs {     notification\_email {       test@163.com     }     notification\_email\_from keepalived@localhost     smtp\_server 127.0.0.1     smtp\_connect\_timeout 30     router\_id LVS\_BACKUP  }  vrrp\_script **chk\_http\_port** {  script "/usr/local/src/check\_nginx\_pid.sh"  interval 2             #（检测脚本执行的间隔）  weight 2  }  vrrp\_instance VI\_1 {      state BACKUP      interface bond0      virtual\_router\_id 51      priority 66      advert\_int 1      authentication {          auth\_type PASS          auth\_pass 1111  }  track\_script {  **chk\_http\_port**            #（调用检测脚本）  }      virtual\_ipaddress {          192.168.15.135      }  } |

以下是针对nginx状态进行检测的脚本，第一次nginx服务死掉时，会重新启动，如果Nginx服务无法正常启动，则杀掉keepalived进程

vim  /usr/local/src/check\_nginx\_pid.sh

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

Ok，开始nginx负载均衡测试，停掉其中一台的任何服务，不影响整个系统的运作｡

# 四．测试

依次启动两个tomcat=》启动两个nginx=》启动两个keepalived，查看主机上是否有虚拟ip 192.168.15.135

通过虚拟ip访问nignx，看页面是不是轮询两个tomcat

将主机的nginx进程kill掉，看看vip是不是切换到备机器上了

访问虚拟ip是不是还能轮询访问tomcat