**网站集群设计及实施方案**

[1. 项目设计方案 2](#_Toc919860860)

[1.1. 需求描述： 2](#_Toc1371248212)

[1.2. 设计方案： 2](#_Toc969608147)

[1.3. 系统设计架构图 3](#_Toc1158185496)

[1.3.1. 现有系统设计架构图 3](#_Toc1158342152)

[1.3.2. 改造后系统设计架构图 3](#_Toc1250108209)

[2. 项目实施方案 4](#_Toc453311554)

[2.1. 需求解决方案描述 4](#_Toc40268899)

[2.2. 环境准备 4](#_Toc67277524)

[2.3. Ip规划 4](#_Toc1213762015)

[2.4. 网站平台的搭建步骤． 5](#_Toc421902513)

[2.4.1. 网站后端平台apache部署 5](#_Toc742588470)

[2.4.1.1. Apache01服务器编写IP与YUM厂库结合脚本 5](#_Toc1048715199)

[2.4.1.2. Apache01服务器配置主机名.IP及yum测试 7](#_Toc1353173115)

[2.4.1.3. Apache01服务器安装http及网站测试 7](#_Toc1867588812)

[2.4.1.4. Apache02与Apache01的配置类似． 8](#_Toc36250453)

[2.4.2. 网站后端平台tomcat部署 8](#_Toc995086243)

[2.4.2.1. tomcat01服务器编写IP与YUM厂库结合脚本 8](#_Toc113965893)

[2.4.2.2. tomcat01服务器配置主机名.IP及yum测试 10](#_Toc1843976242)

[2.4.2.3. Tomcat01服务器安装tomcat及网站测试 10](#_Toc1449991802)

[2.4.2.4. 为tomcat服务器编写测试网页 11](#_Toc609213711)

[2.4.2.5. Tomcat02服务器安装tomcat及网站测试 12](#_Toc1331221462)

[2.4.3. 两台Nginx调速器的配置 12](#_Toc2041123456)

[2.4.3.1. nginx01服务器配置主机名.IP及yum测试 12](#_Toc932731107)

[2.4.3.2. 安装nginx环境所需要的依赖包 13](#_Toc288321494)

[2.4.3.3. 安装nginx 13](#_Toc1363980626)

[2.4.3.4. 修改nginx配置文件，通过location其中网站平台可以将html/css等静态网页，交给后端apache处理．Jsp/php等动态网页交给后端tomcat及php-fpm处理． 13](#_Toc726876930)

[2.4.3.5. 启动nginx调度器的服务及测试 14](#_Toc1554213432)

[2.4.3.6. nginx02服务器配置主机名.IP及yum测试 15](#_Toc1222816331)

[2.4.3.7. 安装nginx和nginx环境所需要的依赖包 15](#_Toc407116293)

[2.4.3.8. 修改nginx配置文件，通过location其中网站平台可以将html/css等静态网页，交给后端apache处理．Jsp/php等动态网页交给后端tomcat及php-fpm处理． 15](#_Toc1058678341)

[2.4.3.9. 启动nginx调度器的服务及测试 16](#_Toc2142677192)

[2.4.4. 负载均衡及高可用LVS+keepalived配置 16](#_Toc1778364505)

# 项目设计方案

## 需求描述：

随着网站平台业务不断扩展，并发访问量的增加，现有平台的架构设计已达到瓶颈，急需新的架构来满足日益增加的需求．

## 设计方案：

旧方案描述：由于机房条件有限，旧的网站平台采用的是一台高性能服务器安装LNMP平台，并将公司网站部署在该平台上，所有访问都是通过直接访问该服务器来实现．

新方案描述：现有机房扩建，购买了大量的服务器，所以新的网站平台设计一共采用７台服务器．其中一台服务器部署kaapalived＋lvs，为网站的负载均衡的功能；网站的前端平台使用２台服务器，并在前端平台使用nginx反向代理，通过location把html．Css.xml等静态页面交给后端apache/nginx处理．把jsp.php等动态页面交给后端tomcat/php-fpm处理，实现网站平台动静分离的效果，提高用户访问静态代码的速度，降低对后台应用访问，从而提高平台的访问速度，达到资源的最大化利用；网站的后端一共使用４台服务器，其中两台部署apache,另外两台部署tomcat．为网站平台提供高效.安全．稳定的业务．

## 系统设计架构图

### 现有系统设计架构图

### 改造后系统设计架构图

# 项目实施方案

## 需求解决方案描述

为了网站高可用和高效的访问需求，网站平台整体分两部分：web前置和后端

1. 一台服务器使用LVS+keepalived为网站前端代理服务器实现负载均衡和高可用功能．
2. Web平台前置使用两台服务做nginx反向代理，并通过修改nginx配置文件中的upstream和location提供集群和网页动静分离的功能．
3. Web平台后端使用四台服务做web集群，其中两台为apache．另外两台为tomcat．保证web的高可用

## 环境准备

服务器系统：contos7.5操作系统／yum厂库准备．

## Ip规划

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **网站web集群平台IP规划** | | | | | | | | | |
| **序号** | **网段** | **业务**IP | **管理**IP | **网关** | **业务** | **使用设备** | **设备名称** | **使用设备**mac地址 | **备注** |
| 1 | 192.168.4.0/24 | 192.168.4.180/24 | 192.168.4.80/24 | 192.168.4.254 | 网站后端平台 | web后端服务器apache01 | apache01 |  |  |
| 2 | 192.168.4.0/24 | 192.168.4.181/24 | 192.168.4.81/24 | 192.168.4.254 | 网站后端平台 | web后端服务器apache02 | apache02 |  |  |
| 3 | 192.168.4.0/24 | 192.168.4.182/24 | 192.168.4.82/24 | 192.168.4.254 | 网站后端平台 | web后端服务器tomcat01 | tomcat01 |  |  |
| 4 | 192.168.4.0/24 | 192.168.4.183/24 | 192.168.4.83/24 | 192.168.4.254 | 网站后端平台 | web后端服务器tomcat02 | tomcat02 |  |  |
| 5 | 192.168.4.0/24 | 192.168.4.184/24 | 192.168.4.84/24 | 192.168.4.254 | 网站前端平台 | web前端服务器nginx01 | nginx-proxy01 |  |  |
| 6 | 192.168.4.0/24 | 192.168.4.185/24 | 192.168.4.85/24 | 192.168.4.254 | 网站前端平台 | web前端服务器nginx02 | nginx-proxy02 |  |  |
| 7 | 192.168.4.0/24 | 192.168.4.186/24 | 192.168.4.86/24 | 192.168.4.254 | 网站平台负载均衡 | web平台服务器keepalived01 | keepalived01 |  |  |

## 网站平台的搭建步骤．

### 网站后端平台apache部署

#### Apache01服务器编写IP与YUM厂库结合脚本

#!/bin/bash

confip(){

ip1=$(echo $ip | awk -F "." '{print $1}')

ip2=$(echo $ip | awk -F "." '{print $2}')

ip3=$(echo $ip | awk -F "." '{print $3}')

eip=${ip1}.${ip2}.${ip3}.254

sip=$(grep baseurl /etc/yum.repos.d/local.repo | awk -F "/" '{print $3}') &> /dev/null

sed -i "/baseurl/s/${sip}/${eip}/" /etc/yum.repos.d/local.repo

}

network(){

if [ -z $gw ];then

nmcli connection modify $name ipv4.method manual ipv4.addresses $ip connection.autoconnect yes &> /dev/null

if [ $? -ne 0 ];then

echo 'Network Address Error'

fi

nmcli connection up $name

confip

else

nmcli connection modify $name ipv4.method manual ipv4.addresses $ip ipv4.gateway $gw connection.autoconnect yes &> /dev/null

if [ $? -ne 0 ];then

echo 'Network Address Error'

fi

nmcli connection up $name

confip

fi

}

read -p 'Network name(eth0/eth1/eth2/eth3):' name

read -p 'Set IP(IP/24):' ip

read -p 'Set Gateway(default none):' gw

if [ -z $name ] || [ -z $ip ];then

echo -e "\e[31;1mPlease enter parameters\e[0m"

exit

fi

echo $ip | grep /24

if [ $? -ne 0 ];then

echo -e "\e[31;1mNetwork Address Error\e[0m"

exit

fi

case $name in

eth0)

network;;

eth1)

network;;

eth2)

network;;

eth3)

network;;

\*)

echo -e "\e[31;1mNetwork Address Error\e[0m"

esac

#### Apache01服务器配置主机名.IP及yum测试

[root@localhost ~]# setip

Network name(eth0/eth1/eth2/eth3):eth0

Set IP(IP/24):192.168.4.180/24

Set Gateway(default none):

192.168.4.180/24

[root@localhost ~]# yum clean all

[root@localhost ~]# yum repolist

Loaded plugins: fastestmirror

Determining fastest mirrors

local\_repo | 3.6 kB 00:00

(1/2): local\_repo/group\_gz | 166 kB 00:00

(2/2): local\_repo/primary\_db | 5.9 MB 00:00

repo id repo name status

local\_repo CentOS-7 - Base 9,911

repolist: 9,911

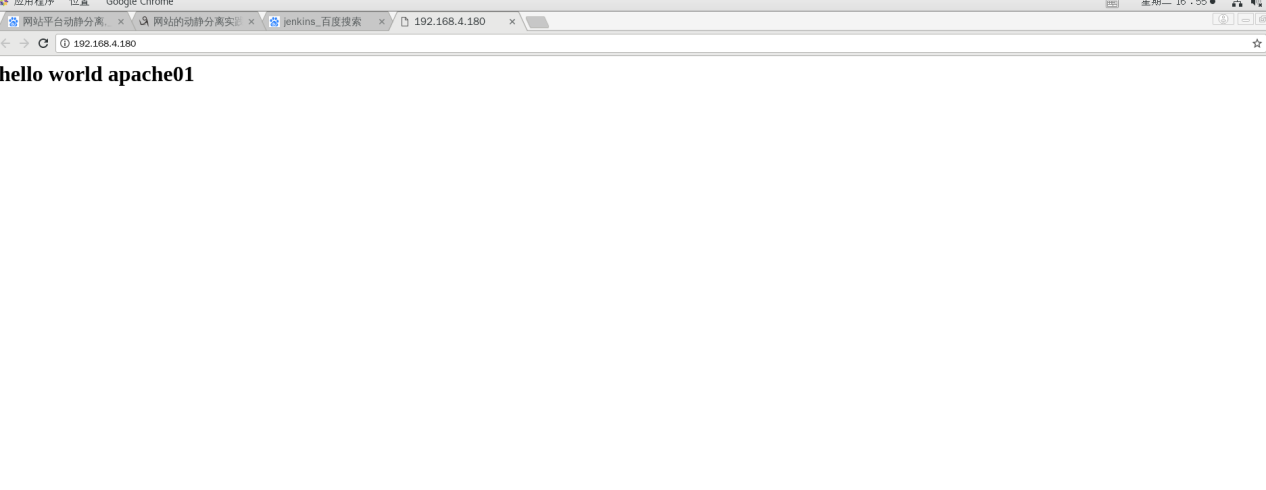
#### Apache01服务器安装http及网站测试

[root@localhost ~]# yum -y install httpd

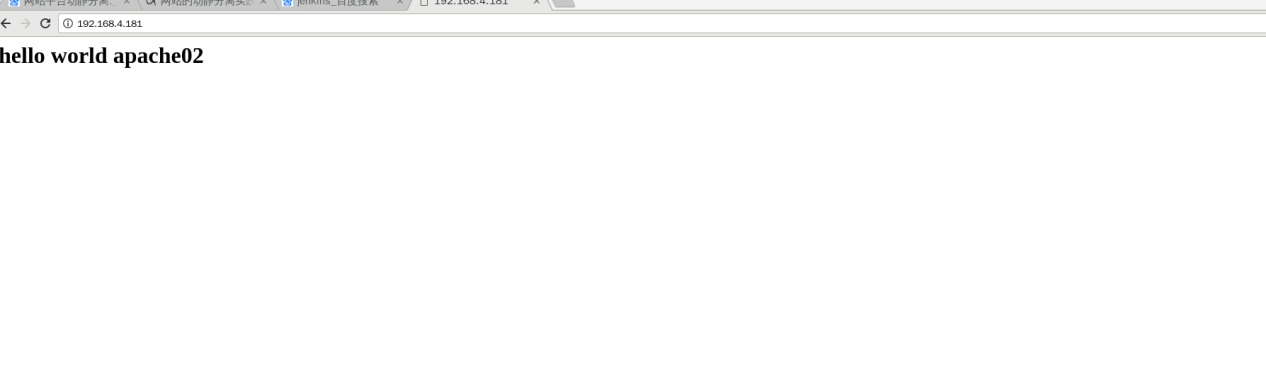
[root@localhost ~]# systemctl start httpd

[root@localhost ~]# systemctl enable httpd

[root@localhost ~]# echo '<h1> hello world apache01</h1>' > /var/www/html/index.html



#### Apache02与Apache01的配置类似．



### 网站后端平台tomcat部署

#### tomcat01服务器编写IP与YUM厂库结合脚本

#!/bin/bash

confip(){

ip1=$(echo $ip | awk -F "." '{print $1}')

ip2=$(echo $ip | awk -F "." '{print $2}')

ip3=$(echo $ip | awk -F "." '{print $3}')

eip=${ip1}.${ip2}.${ip3}.254

sip=$(grep baseurl /etc/yum.repos.d/local.repo | awk -F "/" '{print $3}') &> /dev/null

sed -i "/baseurl/s/${sip}/${eip}/" /etc/yum.repos.d/local.repo

}

network(){

if [ -z $gw ];then

nmcli connection modify $name ipv4.method manual ipv4.addresses $ip connection.autoconnect yes &> /dev/null

if [ $? -ne 0 ];then

echo 'Network Address Error'

fi

nmcli connection up $name

confip

else

nmcli connection modify $name ipv4.method manual ipv4.addresses $ip ipv4.gateway $gw connection.autoconnect yes &> /dev/null

if [ $? -ne 0 ];then

echo 'Network Address Error'

fi

nmcli connection up $name

confip

fi

}

read -p 'Network name(eth0/eth1/eth2/eth3):' name

read -p 'Set IP(IP/24):' ip

read -p 'Set Gateway(default none):' gw

if [ -z $name ] || [ -z $ip ];then

echo -e "\e[31;1mPlease enter parameters\e[0m"

exit

fi

echo $ip | grep /24

if [ $? -ne 0 ];then

echo -e "\e[31;1mNetwork Address Error\e[0m"

exit

fi

case $name in

eth0)

network;;

eth1)

network;;

eth2)

network;;

eth3)

network;;

\*)

echo -e "\e[31;1mNetwork Address Error\e[0m"

esac

#### tomcat01服务器配置主机名.IP及yum测试

[root@localhost ~]# hostnamectl set-hostname tomcat01

[root@localhost ~]# setip

Network name(eth0/eth1/eth2/eth3):eth0

Set IP(IP/24):192.168.4.182/24

Set Gateway(default none):

192.168.4.182/24

[root@localhost ~]# yum clean all

[root@localhost ~]# yum repolist

Loaded plugins: fastestmirror

Determining fastest mirrors

local\_repo | 3.6 kB 00:00

(1/2): local\_repo/group\_gz | 166 kB 00:00

(2/2): local\_repo/primary\_db | 5.9 MB 00:00

repo id repo name status

local\_repo CentOS-7 - Base 9,911

repolist: 9,911

#### Tomcat01服务器安装tomcat及网站测试

https://mirrors.cnnic.cn/apache/tomcat/tomcat-9/v9.0.22/bin/

1. 下载tomcat软件

[root@tomcat01 ~]# wget https://mirrors.cnnic.cn/apache/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz

1. 安装tomcat软件环境需要的依赖包 java-1.8.0-openjdk-devel

[root@tomcat01 tomcat]# yum -y install java-1.8.0-openjdk-devel

1. 给已下载好的tomcat软件包解压，并将其移动到/usr/local/tomcat目录下

[root@tomcat01 ~]#cd

[root@tomcat01 ~]# mv apache-tomcat-9.0.22 /urs/local/tomcat

1. 启动服务，并查看服务状态并测试访问．

[root@tomcat01 tomcat]# /usr/local/tomcat/bin/startup.sh

[root@tomcat01 tomcat]# ss -nltup

Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port

udp UNCONN 0 0 127.0.0.1:323 \*:\* users:(("chronyd",pid=406,fd=1))

tcp LISTEN 0 128 \*:22 \*:\* users:(("sshd",pid=804,fd=3))

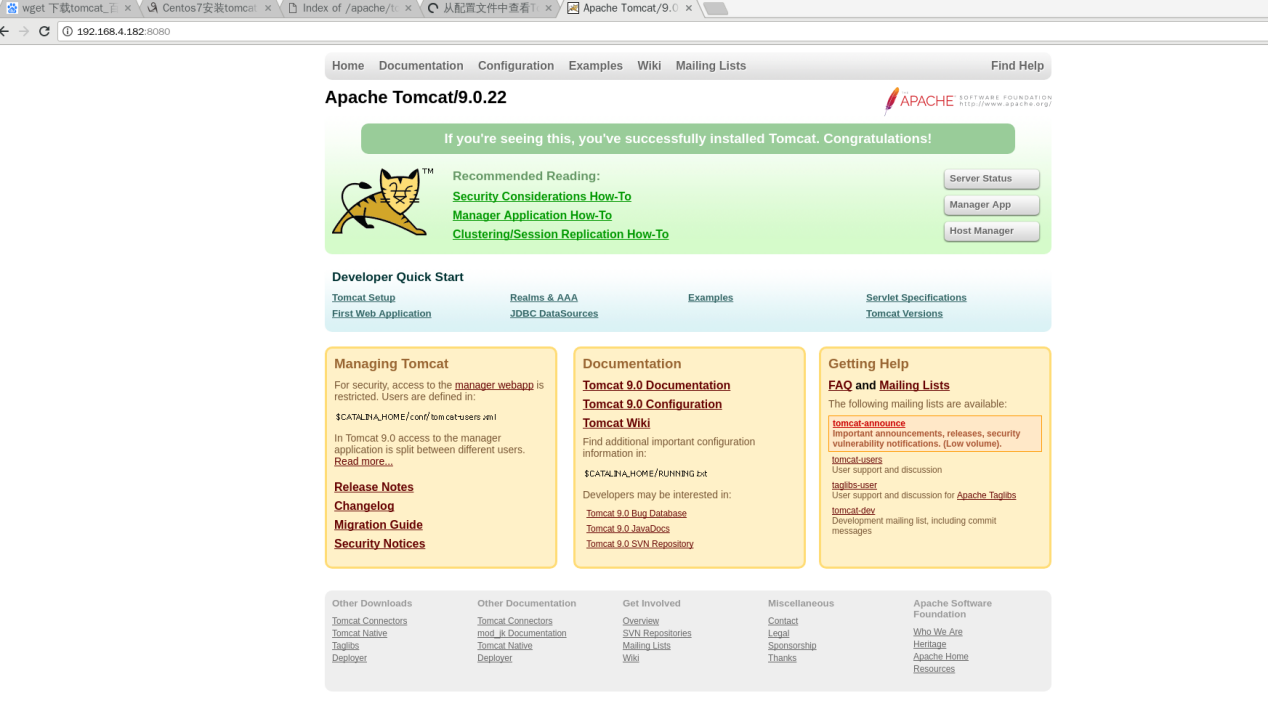
tcp LISTEN 0 1 ::ffff:127.0.0.1:8005 :::\* users:(("java",pid=1054,fd=73))

tcp LISTEN 0 100 :::8009 :::\* users:(("java",pid=1054,fd=59))

tcp LISTEN 0 100 :::8080 :::\* users:(("java",pid=1054,fd=54))

第一个连接器监听8080端口，负责建立HTTP连接。在通过浏览器访问Tomcat服务器的Web应用时，使用的就是这个连接器。

第二个连接器监听8009端口，负责和其他的HTTP服务器建立连接。在把Tomcat与其他HTTP服务器集成时，就需要用到这个连接器．



#### 为tomcat服务器编写测试网页

[root@tomcat01 ROOT]# vim /usr/local/tomcat/webapps/ROOT/test.jsp

<html>

<body>

<center>

Now timeis:<%=new java.util.Date()%>

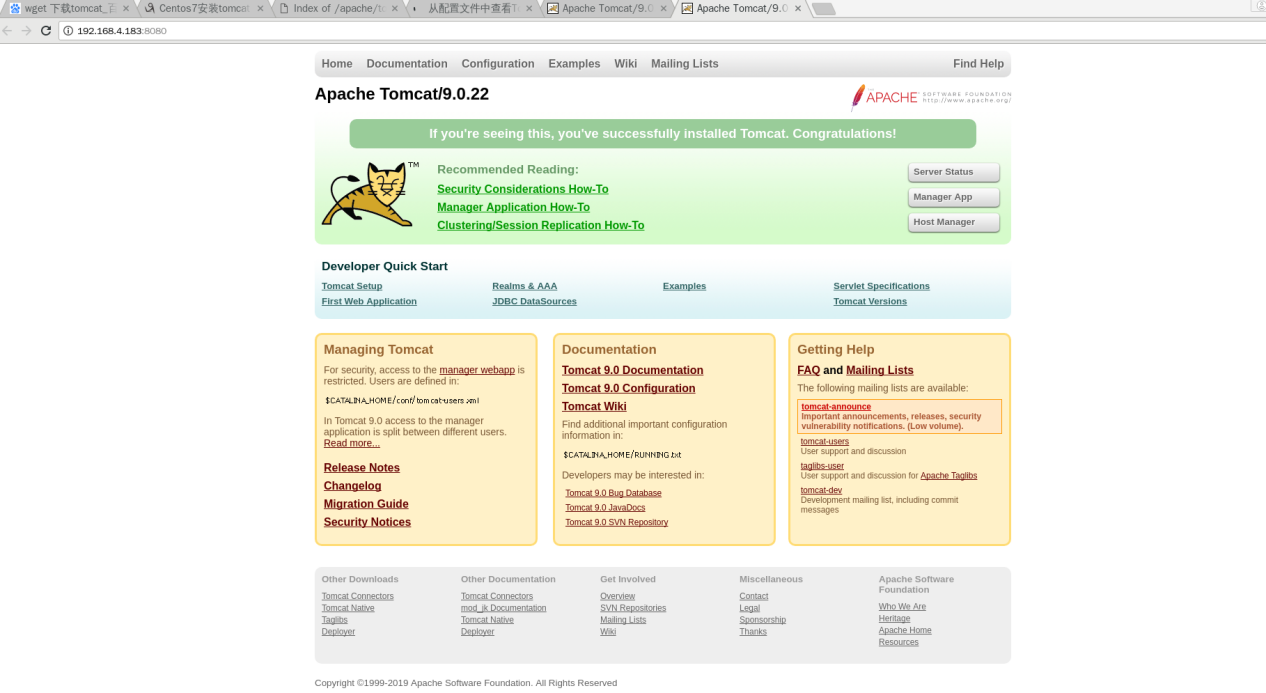
<h1>192.168.4.182</h1>

</center>

</body>

</html>

#### Tomcat02服务器安装tomcat及网站测试



### 两台Nginx调速器的配置

http://nginx.org/download/

#### nginx01服务器配置主机名.IP及yum测试

[root@localhost ~]# hostnamectl set-hostname nginx-proxy01

[root@localhost ~]# setip

Network name(eth0/eth1/eth2/eth3):eth0

Set IP(IP/24):192.168.4.184/24

Set Gateway(default none):

192.168.4.184/24

[root@localhost ~]# yum clean all

[root@localhost ~]# yum repolist

Loaded plugins: fastestmirror

Determining fastest mirrors

local\_repo | 3.6 kB 00:00

(1/2): local\_repo/group\_gz | 166 kB 00:00

(2/2): local\_repo/primary\_db | 5.9 MB 00:00

repo id repo name status

local\_repo CentOS-7 - Base 9,911

repolist: 9,911

#### 安装nginx环境所需要的依赖包

[root@localhost ~]# yum -y install pcre-devel openssl-devel gcc zlib-devel

#### 安装nginx

[root@nginx-proxy01 ~]# tar -zxf nginx-1.12.2.tar.gz

[root@nginx-proxy01 ~]# cd nginx-1.12.2/

[root@nginx-proxy01 nginx-1.12.2]# ls

auto CHANGES CHANGES.ru conf configure contrib html LICENSE man README src

1. 源码编译nginx文件，并加载ssl及status状态模块，为方便nginx监控提供方便及后期安全．

[root@nginx-proxy01 nginx-1.12.2]# ./configure --with-http\_ssl\_module --with-http\_stub\_status\_module

[root@nginx-proxy01 nginx-1.12.2]# make && make install

[root@nginx-proxy01 local]# cd /usr/local/nginx/

[root@nginx-proxy01 nginx]# ls

conf html logs sbin

#### 修改nginx配置文件，通过location其中网站平台可以将html/css等静态网页，交给后端apache处理．Jsp/php等动态网页交给后端tomcat及php-fpm处理．

定义app1web集群处理网站平台的静态请求，app2web集群处理网站平台的动态请求，从而提供网站的访问速度和高可用需求．nginx配置文件如下：

...

upstream app1web{

server 192.168.4.180:80;

server 192.168.4.181:80;

}

upstream app2web{

server 192.168.4.182:8080;

server 192.168.4.183:8080;

}

...

location ~ .\*\.(html|js|css|gif|jpg|jpeg|png)$ {

proxy\_pass http://app1web;

}

location ~ .\*\.(jsp|php)$ {

proxy\_pass http://app2web;

}

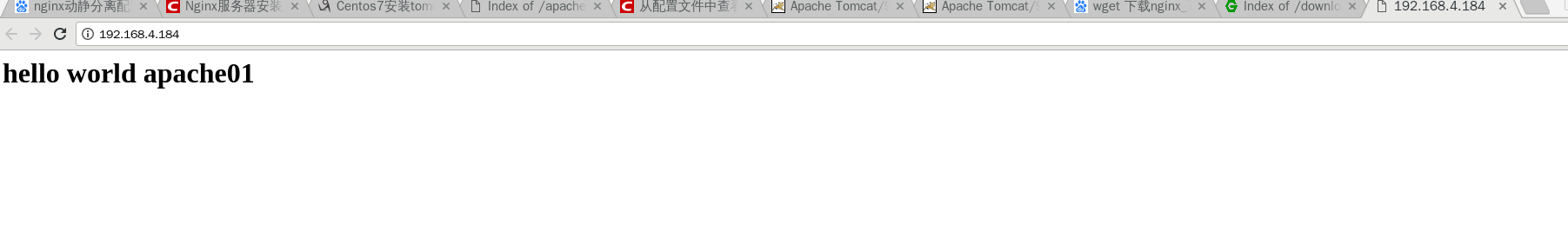
...

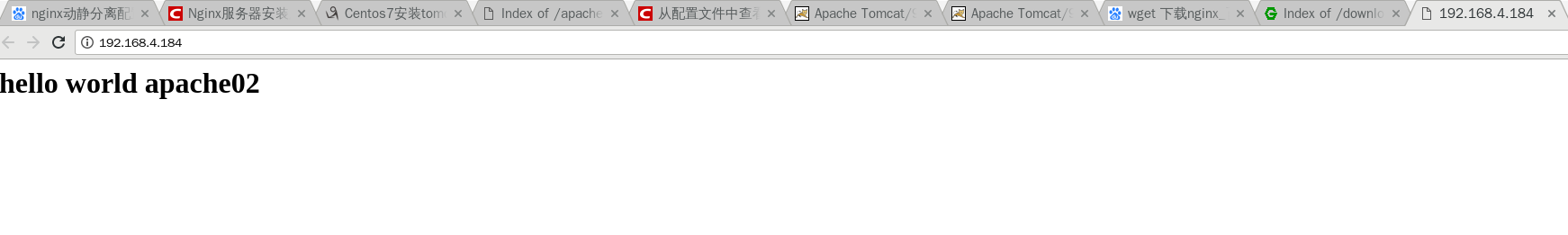
#### 启动nginx调度器的服务及测试

[root@nginx-proxy01 conf]# ln -s /usr/local/nginx/sbin/nginx /usr/sbin/

[root@nginx-proxy01 conf]# nginx

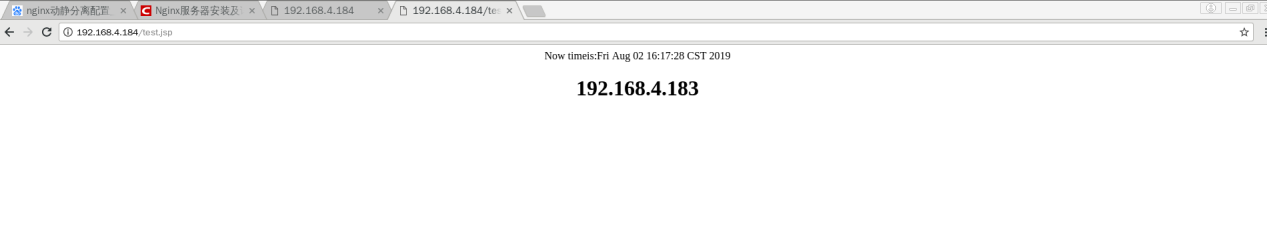
http://192.168.4.184





http://192.168.4.184/tes.jsp





#### nginx02服务器配置主机名.IP及yum测试

[root@localhost ~]# hostnamectl set-hostname nginx-proxy02

[root@localhost ~]# setip

Network name(eth0/eth1/eth2/eth3):eth0

Set IP(IP/24):192.168.4.185/24

Set Gateway(default none):

192.168.4.185/24

[ 7814.881295] IPv6: ADDRCONF(NETDEV\_CHANGE): eth0: link becomes ready

Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)

#### 安装nginx和nginx环境所需要的依赖包

[root@localhost ~]# yum -y install pcre-devel gcc openssl-devel zlib-devel

[root@nginx-proxy02 nginx-1.12.2]# ./configure --with-http\_ssl\_module --with-http\_stub\_status\_module

[root@nginx-proxy02 nginx-1.12.2]# make && make install

#### 修改nginx配置文件，通过location其中网站平台可以将html/css等静态网页，交给后端apache处理．Jsp/php等动态网页交给后端tomcat及php-fpm处理．

定义app1web集群处理网站平台的静态请求，app2web集群处理网站平台的动态请求，从而提供网站的访问速度和高可用需求．nginx配置文件如下：

...

upstream app1web{

server 192.168.4.180:80;

server 192.168.4.181:80;

}

upstream app2web{

server 192.168.4.182:8080;

server 192.168.4.183:8080;

}

...

location ~ .\*\.(html|js|css|gif|jpg|jpeg|png)$ {

proxy\_pass http://app1web;

}

location ~ .\*\.(jsp|php)$ {

proxy\_pass http://app2web;

}

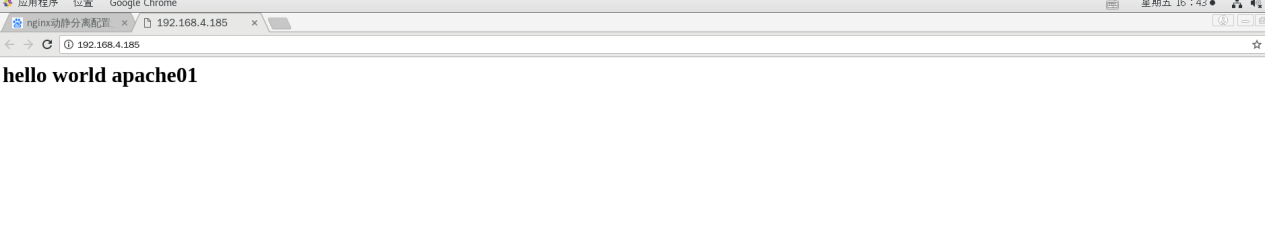
...

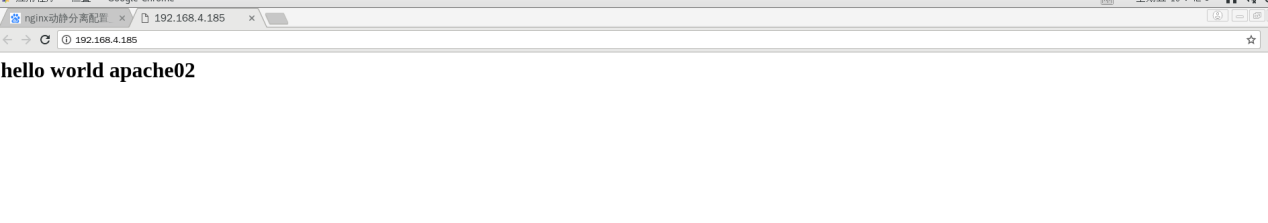
#### 启动nginx调度器的服务及测试

[root@nginx-proxy02 nginx-1.12.2]# ln -s /usr/local/nginx/sbin/nginx /usr/sbin/

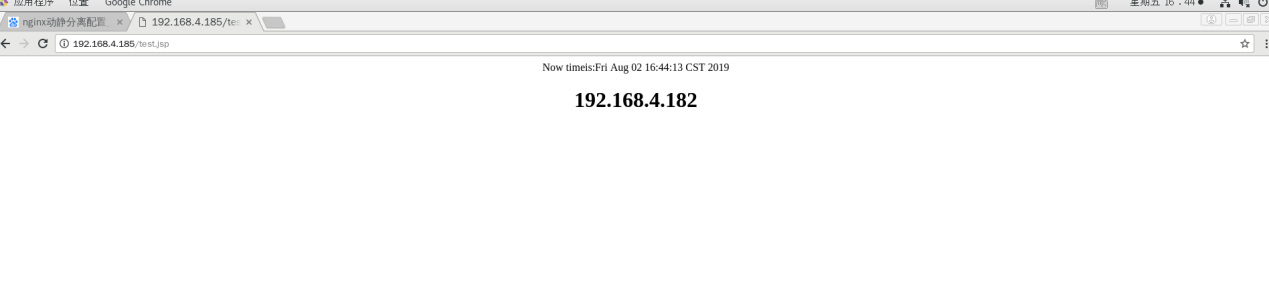
[root@nginx-proxy02 nginx-1.12.2]# nginx

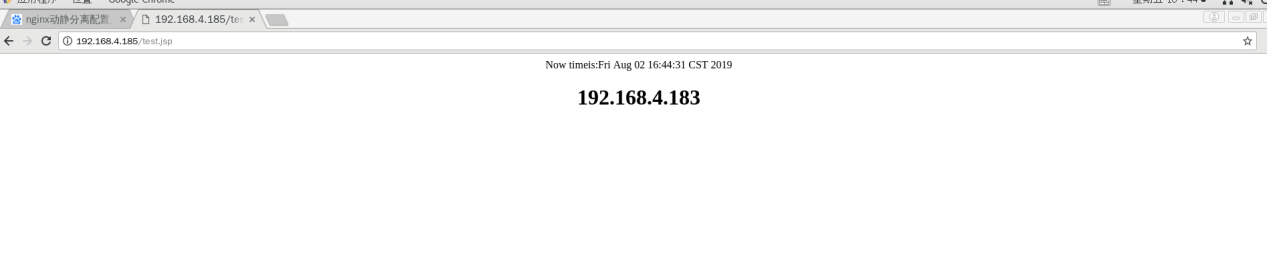
http://192.168.4.185





http://192.168.4.185/test.jsp





### 负载均衡及高可用LVS+keepalived配置