第三十章：Nginx+lamp构建动静分离项目

**一、nginx代理的概述；**

**二、nginx之正向代理；**

**三、nginx之反向代理；**

**四、nginx之负载均衡；**

**五、Nginx之会话session；**

**六、案例：结合nginx反向代理构建nginx+lamp实现动静页面分离以及php服务器的负载均衡；**

**一、nginx代理的概述；**

**概述：**nginx是一款自由的、开源的、高性能的HTTP服务器和反向代理服务器；同时也是一个IMAP、POP3、SMTP代理服务器；nginx可以作为一个内部网络代理上网的代理服务器，另外nginx可以作为反向代理进行负载均衡的实现。

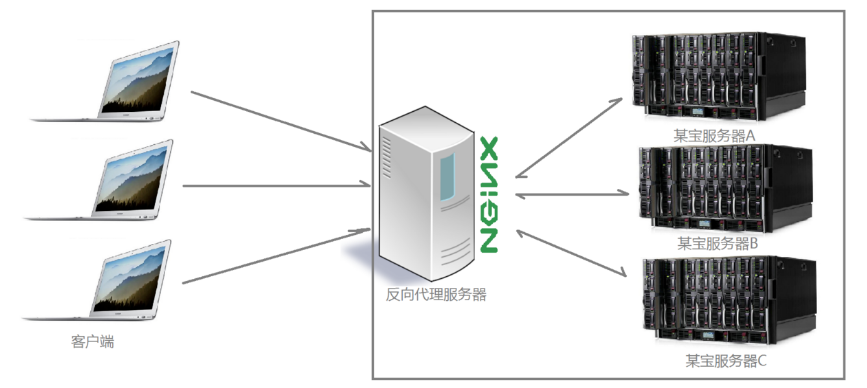
**二、nginx之正向代理；**





**解析：**在如今的网络环境下，我们如果由于技术需要要去访问国外的某些网站，此时你会发现位于国外的某网站我们通过浏览器是没有办法访问的，此时大家可能都会用一个操作FQ进行访问，FQ的方式主要是找到一个可以访问国外网站的代理服务器，我们将请求发送给代理服务器，代理服务器去访问国外的网站，然后将访问到的数据传递给我们！上述这样的代理模式称为正向代理，正向代理最大的特点是客户端非常明确要访问的服务器地址；服务器只清楚请求来自哪个代理服务器，而不清楚来自哪个具体的客户端；正向代理模式屏蔽或者隐藏了真实客户端信息。

**三、nginx之反向代理；**



**第一层：**client，客户端层，发出http请求的单位；

**第二层：**反向代理层，agent层，负责将client客户端的请求，代理转发到第三层（real server）真实的服务器的节点，并且保证real server节点不暴露在client前；

**第三层：**real server，服务器层，真实负责处理cleint客户端的请求，通常在此为多台服务器构成，形成负载均衡之效果；

**四、nginx之负载均衡；**

**概述：**nginx扮演了反向代理服务器的角色，它是以依据什么样的规则进行请求分发的呢？不用的项目应用场景，分发的规则是否可以控制呢？

**实现方式：**

硬件负载均衡：也称为硬负载，如F5负载均衡，相对造价昂贵成本较高，但是数据的稳定性安全性等等有非常好的保障，如中国移动中国联通这样的公司才会选择硬负载进行操作；

软件负载均衡：更多的公司考虑到成本原因，会选择使用软件负载均衡，软件负载均衡是利用现有的技术结合主机硬件实现的一种消息队列分发机制；

**nginx的负载均衡算法：**

1.weight轮询（默认）：接收到的请求按照顺序逐一分配到不同的后端服务器。这种方式下，可以给不同的后端服务器设置一个权重值（weight），用于调整不同的服务器上请求的分配率；权重数据越大，被分配到请求的几率越大；

2.ip\_hash：每个请求按照发起客户端的ip的hash结果进行匹配，这样的算法下一个固定ip地址的客户端总会访问到同一个后端服务器，这也在一定程度上解决了集群部署环境下session共享的问题；

3.fair：智能调整调度算法，动态的根据后端服务器的请求处理到响应的时间进行均衡分配，响应时间短处理效率高的服务器分配到请求的概率高，响应时间长处理效率低的服务器分配到的请求少；结合了前两者的优点的一种调度算法。但是需要注意的是nginx默认不支持fair算法，如果要使用这种调度算法，请安装upstream\_fair模块；

4.url\_hash：按照访问的url的hash结果分配请求，每个请求的url会指向后端固定的某个服务器，可以在nginx作为静态服务器的情况下提高缓存效率。同样要注意nginx默认不支持这种调度算法，要使用的话需要安装nginx的hash软件包；

5.最小连接数法：最小连接数算法比较灵活和智能，由于后端服务器的配置不尽相同，对于请求的处理有快有慢，它是根据后端服务器当前的连接情况，动态地选取其中当前积压连接数最少的一台服务器来处理当前的请求，尽可能地提高后端服务的利用效率，将负责合理地分流到每一台服务器。

**五、Nginx之会话session：**

**概述：**在上述负载均衡中，real server多个web节点的服务器状态必须统一，如session会话状态，会话保持是指在负载均衡器上有一种机制，在作负载均衡的同时，还保证同一用户相关连的访问请求会被分配到同一台服务器上；

**举例理解会话保持：**  
1.如果有一个用户访问请求被分配到服务器A，并且在服务器A登录了，并且在很短的时间，这个用户又发出了一个请求，如果没有会话保持功能的话，这个用户的请求很有可能会被分配到服务器B去，这个时候在服务器B上是没有登录的，所以你要重新登录，但是用户并不知道自己的请求被分配到了哪里，用户的感觉就是登录了，怎么又要登录，用户体验很不好；  
2.你在淘宝上面买东西，从登录＝》拍得东西＝》添加地址＝》付款，这是一个一系列的过程，也可以理解成一次操作过程，所有这一系列的操作过程都应当由一台服务器完成，而不能被负载均衡器分配到不同的服务器上；

**会话保持实现方式：**

1.调整负载调度算法为ip\_hash；ip\_hash使用源地址哈希算法，将同一客户端的请求总是发往同一个后端服务器，除非该服务器不可用。

当后端服务器宕机后，session会丢失；

来自同一局域网的客户端会被转发到同一个后端服务器，可能导致负载失衡；

不适用于CDN网络，不适用于前段还有代理的情况。

2. nginx模块实现：sticky\_cookie\_insert：sticky\_cookie\_insert启用会话亲缘关系，这会导致来自同一客户端的请求被传递到一组服务器在同一台服务器。与ip\_hash不同之处在于，它不是基于IP来判断客户端的，而是基于cookie来判断。因此可以避免上述ip\_hash中来自同一局域网的客户端和前段代理导致负载失衡的情况。

**会话保持限制：**会话保持都会有时间的限制（映射到固定某一台的服务器除外，如：ip\_hash）,各种负载均衡工具都会提供这种会话保持时间的设置，LVS，apache等。连php语言都提供了会话保持时间的设定session.gc\_maxlifetime会话保持时间的设定要大于session生存时间的设定，这样可以减少需要同步session的情况，但是不能杜绝。所以同步session还是要做的；

**会话同步概述：**在做了web集群后，你肯定会首先考虑session同步问题，因为通过负载均衡后，同一个IP访问同一个页面会被分配到不同的服务器上，如果session不同步的话，一个登录用户，一会是登录状态，一会又不是登录状态。

**会话同步方式：**

1、利用数据库同步session ：  
a，用一个低端电脑建个数据库专门存放web服务器的session，或者，把这个专门的数据库建在文件服务器上，用户访问web服务器时，会去这个专门的数据库check一下session的情况，以达到session同步的目的。   
b，这种方法是把存放session的表和其他数据库表放在一起，如果mysql也做了集群了话，每个mysql节点都要有这张表，并且这张session表的数据表要实时同步。   
说明：用数据库来同步session，会加大数据库的负担，数据库本来就是容易产生瓶颈的地方，如果把session还放到数据库里面，无疑是雪上加霜。上面的二种方法，第一点方法较好，把放session的表独立开来，减轻了真正数据库的负担

2、利用cookie同步session ：  
session是文件的形势存放在服务器端的，cookie是文件的形势存在客户端的，怎么实现同步呢？方法很简单，就是把用户访问页面产生的session放到cookie里面，就是以cookie为中转站。你访问web服务器A，产生了session把它放到cookie里面了，你访问被分配到web服务器B，这个时候，web服务器B先判断服务器有没有这个session，如果没有，在去看看客户端的cookie里面有没有这个session，如果也没有，说明session真的不存，如果cookie里面有，就把cookie里面的sessoin同步到web服务器B，这样就可以实现session的同步了。

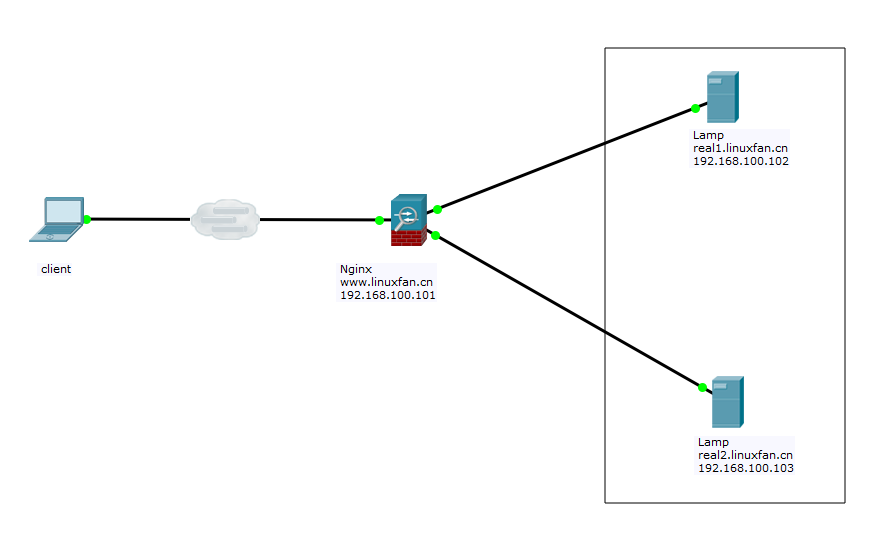
说明：这种方法实现起来简单，方便，也不会加大数据库的负担，但是如果客户端把cookie禁掉了的话，那么session就无从同步了，这样会给网站带来损失；cookie的安全性不高，虽然它已经加了密，但是还是可以伪造的。

3、利用memcache同步session ：  
memcache可以做分布式，如果没有这功能，他也不能用来做session同步。他可以把web服务器中的内存组合起来，成为一个"内存池"，不管是哪个服务器产生的sessoin都可以放到这个"内存池"中，其他的都可以使用。

优点：以这种方式来同步session，不会加大数据库的负担，并且安全性比用cookie大大的提高，把session放到内存里面，比从文件中读取要快很多，推荐使用此种方式。   
缺点：memcache把内存分成很多种规格的存储块，有块就有大小，这种方式也就决定了，memcache不能完全利用内存，会产生内存碎片，如果存储块不足，还会产生内存溢出。

**六、案例：结合nginx反向代理构建nginx+lamp实现动静页面分离以及php服务器的负载均衡**

**案例拓扑：**



**案例环境：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 内存 | 所需软件 |
| Centos 7.4 1708 64bit | 192.168.100.101 | www.linuxfan.cn | 1G | nginx-1.12.2.tar.gz |
| Centos 6.5 64bit | 192.168.100.150 | real1.linuxfan.cn | 1G | httpd-2.2.17.tar.gz、  cmake-2.8.6.tar.gz、  mysql-5.5.22.tar.gz、  libmcrypt-2.5.8.tar.gz、  mcrypt-2.6.8.tar.gz、  mhash-0.9.9.9.tar.gz、  php-5.3.28.tar.gz、  ZendGuardLoader-php-5.3-linux-glibc23-x86\_64.tar.gz |
| Centos 6.5 64bit | 192.168.100.151 | real2.linuxfan.cn | 1G | httpd-2.2.17.tar.gz、  cmake-2.8.6.tar.gz、  mysql-5.5.22.tar.gz、  libmcrypt-2.5.8.tar.gz、  mcrypt-2.6.8.tar.gz、  mhash-0.9.9.9.tar.gz、  php-5.3.28.tar.gz、  ZendGuardLoader-php-5.3-linux-glibc23-x86\_64.tar.gz |

**案例步骤：**

* 部署real1.linuxfan.cn节点的lamp服务；
* 部署real2.linuxfan.cn节点的lamp服务；
* 安装www.linuxfan.cn节点的nginx服务；
* 配置nginx节点的反向代理服务；
* 客户端访问测试代理集群；
* 查看日志分析访问情况；
* 调整负载调度算法；
* 配置Nginx会话session保持；
* **部署real1.linuxfan.cn节点的lamp服务；**

[root@real1 ~]# ls

lamp\_install\_publis-app-2015-07-16.tar.xz

[root@real1 ~]# tar Jxvf lamp\_install\_publis-app-2015-07-16.tar.xz

[root@real1 ~]# cd bin/

[root@real1 bin]#./apache\_install.sh &&./mysql\_install.sh &&./mysql\_config.sh &&./php\_install.sh &&./php\_config.sh &&./lamp\_config.sh

[root@real1 bin]# cd

[root@real1 ~]# netstat -utpln |grep 80

tcp 0 0 :::80 :::\* LISTEN 50971/httpd

[root@real1 ~]# netstat -utpln |grep 3306

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 6693/mysqld

[root@real1 ~]# rm -rf /usr/local/httpd/htdocs/index.html

[root@real1 ~]# vi /usr/local/httpd/htdocs/index.php

<?php

session\_start();

$\_SESSION['time'] =date("Y:m:d:H:s",time());

echo "本次访问时间"."<font color=red>".$\_SESSION['time']."</font>"."<br>";

echo "访问的服务器地址是"."<font color=red>".$\_SERVER['SERVER\_ADDR']."</font>"."<br>";

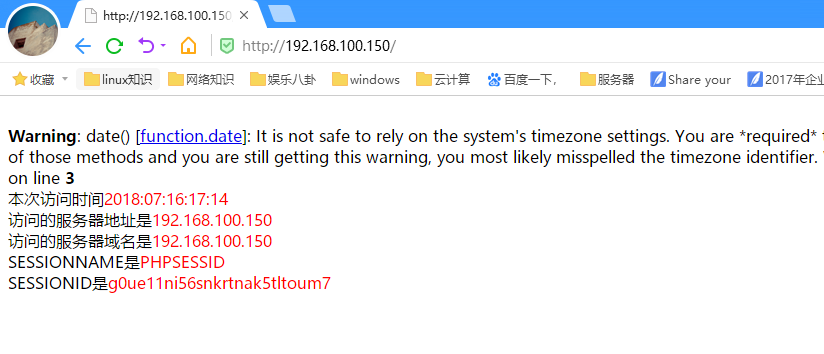
echo "访问的服务器域名是"."<font color=red>".$\_SERVER['SERVER\_NAME']."</font>"."<br>";

echo "SESSIONNAME是"."<font color=red>".session\_name()."</font>"."<br>";

echo "SESSIONID是"."<font color=red>".session\_id()."</font>"."<br>";

?>

客户端访问测试：



* **部署real2.linuxfan.cn节点的lamp服务；**

[root@real2 ~]# ls

lamp\_install\_publis-app-2015-07-16.tar.xz

[root@real2 ~]# tar Jxvf lamp\_install\_publis-app-2015-07-16.tar.xz

[root@real2 ~]# cd bin/

[root@real2 bin]#./apache\_install.sh &&./mysql\_install.sh &&./mysql\_config.sh &&./php\_install.sh &&./php\_config.sh &&./lamp\_config.sh

[root@real2 bin]# cd

[root@real2 ~]# netstat -utpln |grep 80

tcp 0 0 :::80 :::\* LISTEN 50971/httpd

[root@real2 ~]# netstat -utpln |grep 3306

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 6693/mysqld

[root@real2 ~]# rm -rf /usr/local/httpd/htdocs/index.html

[root@real2 ~]# vi /usr/local/httpd/htdocs/index.php

<?php

session\_start();

$\_SESSION['time'] =date("Y:m:d:H:s",time());

echo "本次访问时间"."<font color=red>".$\_SESSION['time']."</font>"."<br>";

echo "访问的服务器地址是"."<font color=red>".$\_SERVER['SERVER\_ADDR']."</font>"."<br>";

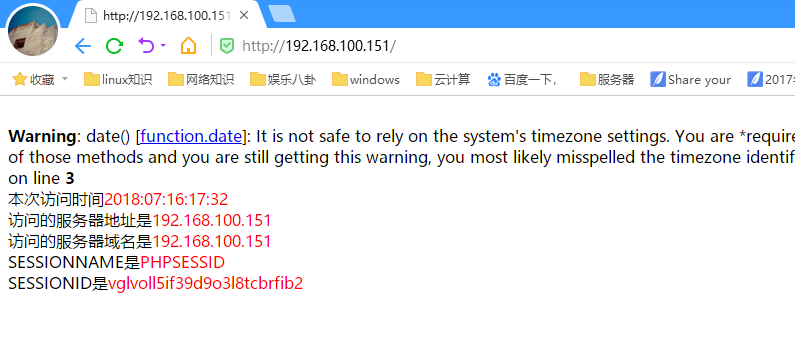
echo "访问的服务器域名是"."<font color=red>".$\_SERVER['SERVER\_NAME']."</font>"."<br>";

echo "SESSIONNAME是"."<font color=red>".session\_name()."</font>"."<br>";

echo "SESSIONID是"."<font color=red>".session\_id()."</font>"."<br>";

?>

客户端访问测试：



* **安装www.linuxfan.cn节点的nginx服务；**

[root@www ~]# hostnamectl set-hostname www.linuxfan.cn

[root@www ~]# hostname

www.linuxfan.cn

[root@www ~]# yum -y install pcre-devel zlib-devel

[root@www ~]# useradd -M -s /sbin/nologin nginx

[root@www ~]# tar zxvf nginx-1.12.2.tar.gz -C /usr/src/

[root@www ~]# cd /usr/src/nginx-1.12.2/

[root@www nginx-1.12.2]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_stub\_status\_module

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

[root@www nginx-1.12.2]# cd

[root@www ~]# ln -s /usr/local/nginx/sbin/nginx /usr/local/sbin/

[root@www ~]# vi /usr/lib/systemd/system/nginx.service

[Unit]

Description=nginxapi

After=network.target

[Service]

Type=forking

PIDFile=/usr/local/nginx/logs/nginx.pid

ExecStart=/usr/local/nginx/sbin/nginx

ExecReload=kill -s HUP $(cat /usr/local/nginx/logs/nginx.pid)

ExecStop=kill -s QUIT $(cat /usr/local/nginx/logs/nginx.pid)

PrivateTmp=Flase

[Install]

WantedBy=multi-user.target

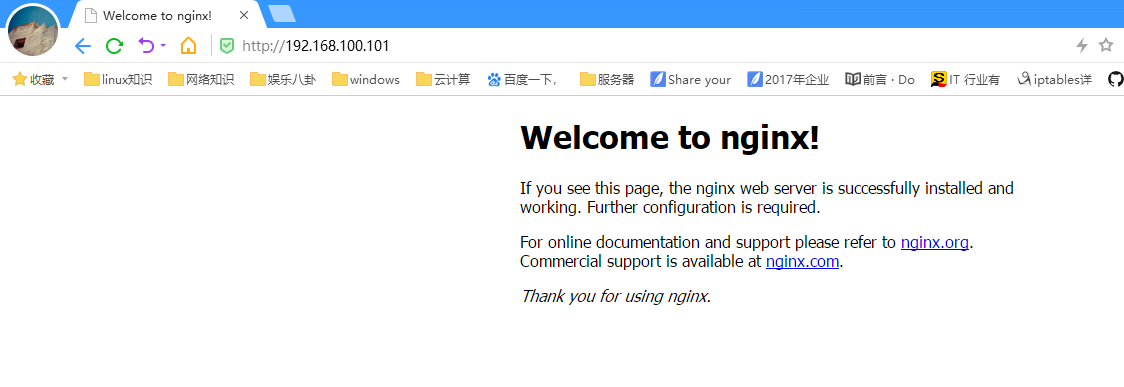
[root@www ~]# systemctl start nginx

[root@www ~]# systemctl enable nginx

[root@www ~]# netstat -utpln |grep nginx

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 3538/nginx: master

客户端访问测试：



* **配置nginx节点的反向代理服务；**

[root@www ~]# vi /usr/local/nginx/conf/nginx.conf

34 upstream lampserver { ##在http区域内，server区域上，名称内不能有下划线

35 server 192.168.100.150:80 weight=1;

36 server 192.168.100.151:80 weight=1;

37 }

47 location / {

48 root html;

49 index index.html index.htm;

50 }

51 location ~ \.(asp|aspx|php|jsp|do|js|css|png|jpg)$ { ##在server区域内添加

52 proxy\_pass http://lampserver;

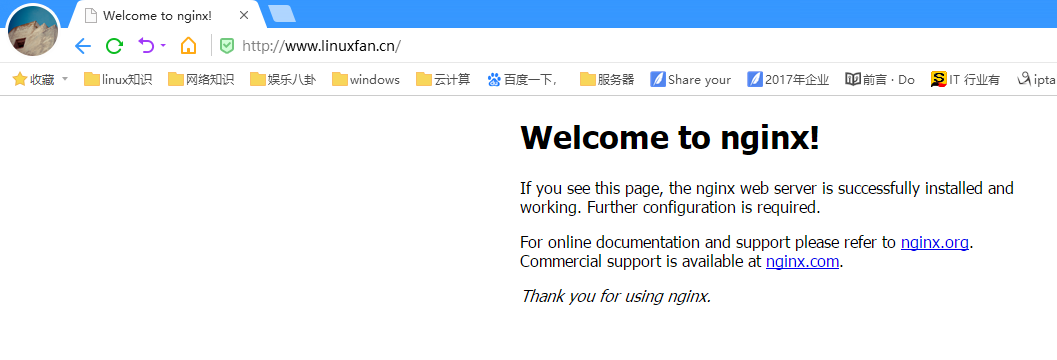
53 }

[root@www ~]# systemctl stop nginx

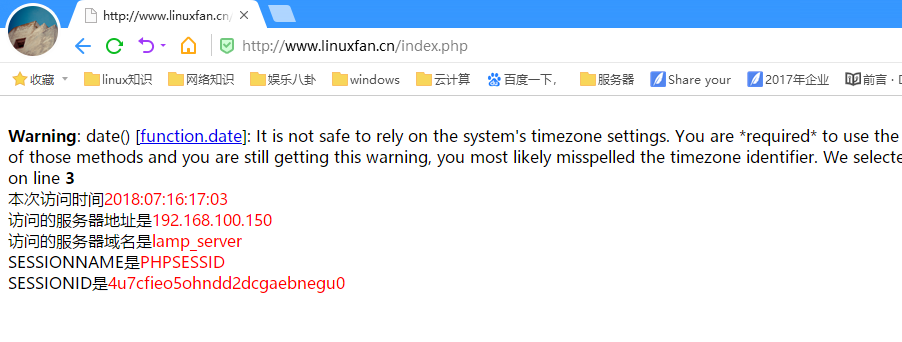
[root@www ~]# systemctl start nginx

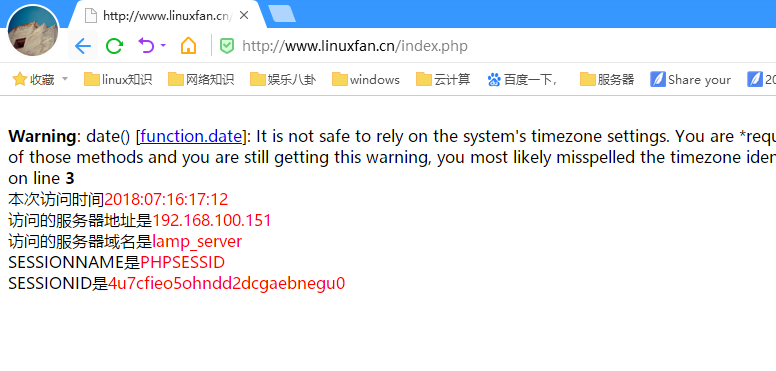
* **客户端访问测试代理集群；**

**客户端访问测试html站点：**



**客户端访问测试php站点：**





* **查看日志分析访问情况；**

[root@www ~]# tail -3 /usr/local/nginx/logs/access.log

192.168.100.1 - - [16/Jul/2018:17:01:34 +0800] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 SE 2.X MetaSr 1.0"

192.168.100.1 - - [16/Jul/2018:17:02:02 +0800] "GET /index.php HTTP/1.1" 200 797 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 SE 2.X MetaSr 1.0"

192.168.100.1 - - [16/Jul/2018:17:02:11 +0800] "GET /index.php HTTP/1.1" 200 797 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 SE 2.X MetaSr 1.0"

[root@real1 ~]# tail -3 /usr/local/httpd/logs/access\_log

192.168.100.102 - - [16/Jul/2018:17:07:55 +0800] "GET /index.php HTTP/1.0" 200 797

192.168.100.102 - - [16/Jul/2018:17:07:56 +0800] "GET /index.php HTTP/1.0" 200 797

192.168.100.102 - - [16/Jul/2018:17:09:03 +0800] "GET /index.php HTTP/1.0" 200 797

[root@real2 ~]# tail -3 /usr/local/httpd/logs/access\_log

192.168.100.102 - - [16/Jul/2018:17:07:56 +0800] "GET /index.php HTTP/1.0" 200 797

192.168.100.102 - - [16/Jul/2018:17:07:56 +0800] "GET /index.php HTTP/1.0" 200 797

192.168.100.102 - - [16/Jul/2018:17:09:12 +0800] "GET /index.php HTTP/1.0" 200 797

* 调整负载调度算法；

[root@www ~]# vi /usr/local/nginx/conf/nginx.conf

upstream lamp\_server {

#ip\_hash; ##ip\_hash算法

#least\_conn; ##最小连接算法

server 192.168.100.150:80 weight=1 down backup max\_fails=3 fail\_timeout=30;

server 192.168.100.151:80 weight=1;

}

注解：

down ：当前节点不参与负载；

backup ：备份节点，当其他节点出现down，该节点才会被访问；

max\_fails=3 ：节点的最大失败次数；

fail\_timeout=30 ：当节点达到最大失败次数，暂停节点的时间；

* **配置Nginx会话session保持；**

**会话保持：**

**方式一：ip\_hash调度算法**

[root@www ~]# vi /usr/local/nginx/conf/nginx.conf

34 upstream lamp\_server {

35 ip\_hash;

36 server 192.168.100.150:80 weight=1;

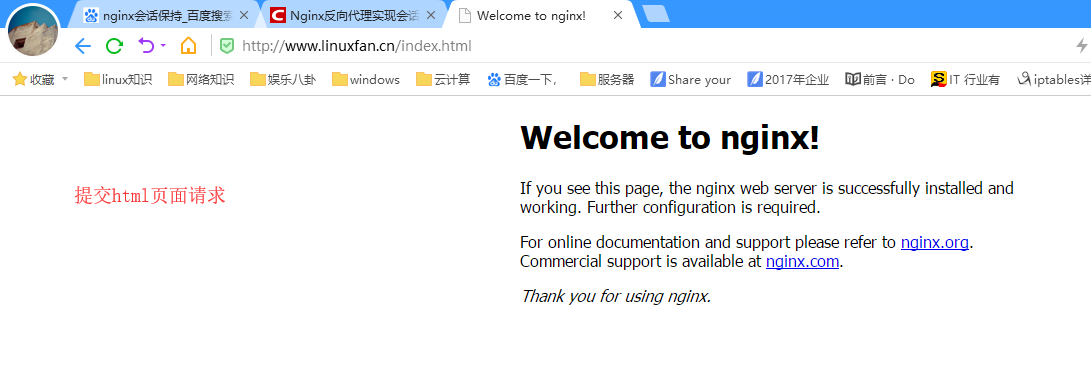
37 server 192.168.100.151:80 weight=1;

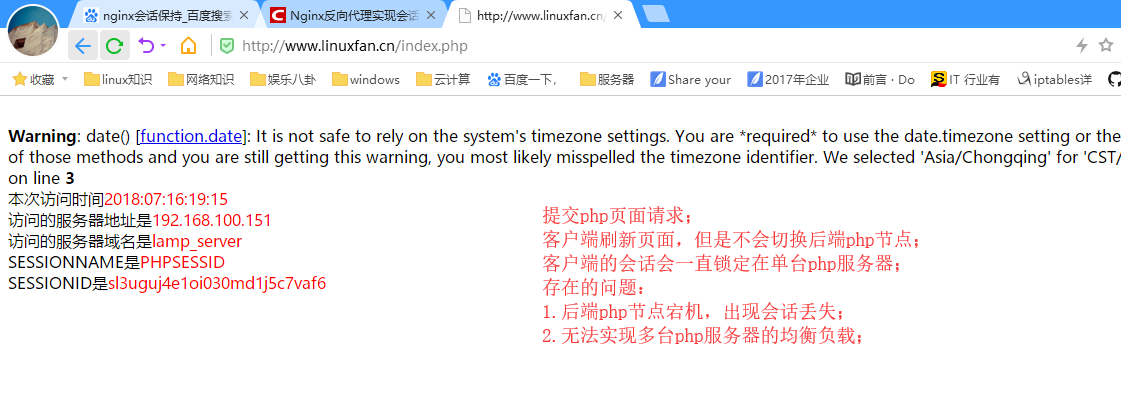
38 }

[root@www ~]# systemctl stop nginx

[root@www ~]# systemctl start nginx

客户端测试：





**方式二：nginx-goodies-nginx-sticky-module-ng-08a395c66e42.zip**

[root@www ~]# ls

nginx-1.6.0.tar.gz nginx-goodies-nginx-sticky-module-ng-08a395c66e42.zip

[root@www ~]# unzip nginx-goodies-nginx-sticky-module-ng-08a395c66e42.zip

[root@www ~]# cp -rf nginx-goodies-nginx-sticky-module-ng-08a395c66e42 /usr/src/

[root@www ~]# tar zxvf nginx-1.6.0.tar.gz -C /usr/src/

[root@www ~]# cd /usr/src/nginx-1.6.0/

[root@www nginx-1.6.0]#./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_stub\_status\_module --add-module=/usr/src/nginx-goodies-nginx-sticky-module-ng-08a395c66e42

[root@www nginx-1.6.0]# make && make install

[root@www nginx-1.6.0]# cd

[root@www ~]# vi /usr/local/nginx/conf/nginx.conf

34 upstream lamp\_server {

35 sticky;

36 server 192.168.100.150:80 weight=1;

37 server 192.168.100.151:80 weight=1;

38 }

[root@www ~]# /usr/local/nginx/sbin/nginx

客户端测试：

