https://blog.csdn.net/wujiangwei567/article/details/77905731

**nginx反向代理深入理解及session共享**

2017年09月09日 09:32:59 [栁罗风尘](https://me.csdn.net/wujiangwei567) 阅读数：985 标签： [nginx](http://so.csdn.net/so/search/s.do?q=nginx&t=blog) 更多

个人分类： [系统运维架构](https://blog.csdn.net/wujiangwei567/article/category/7143144)

**一、反向代理详解**

上次使用nginx做负载均衡，只使用了一个参数weight，详细的参数今天来八一八；

1. upstream newServer{
2. server 192.168.0.1:8081 down;
3. server 192.168.0.2:80 weight=100;
4. server 192.168.0.3:9194;
5. server 192.168.0.3:9194 max\_fails=120 fail\_timeout=12;
6. server 192.168.0.4:8544 backup;
7. }

在需要负载的server节点下添加

proxy\_pass http:*//newServer*

upstream 解释:  
down 表示当前server 不参与负载  
weight 默认为1 数字越大 负载权重越大  
max\_fails 允许请求失败的次数 当超过最大次数 返回proxy\_next\_upstream模块定义的错误  
fail\_timeout:max\_fails次失败后，暂停的时间  
backup: 其他所有的非backup机器down或者忙的时候 请求这台机器，这台机器的压力最小  
  
  
可以同时设置多个upstream 来服务于多个server  
但是一个server只能使用一个 upstream

**二、反向代理sesioon共享问题**

如果使用反向代理，负载均衡，那么就会有个问题，我登录的使用使用的服务器1，而跳转到首页的时候，负载到了

服务器二，而服务器二并没有服务器一的登录信息，又会需要登录，这样可能会出现登录死循环，那么怎么解决

登录信息（session）共享问题呢

方法一.使用redis或者数据库之类的存储session  
  
方法二.负载加入 ip\_hash   
      这样每个请求来的ip会请求固定的一台服务器

我们用[上一次](http://blog.csdn.net/wujiangwei567/article/details/77776254)的例子来测试一下ip\_hahs

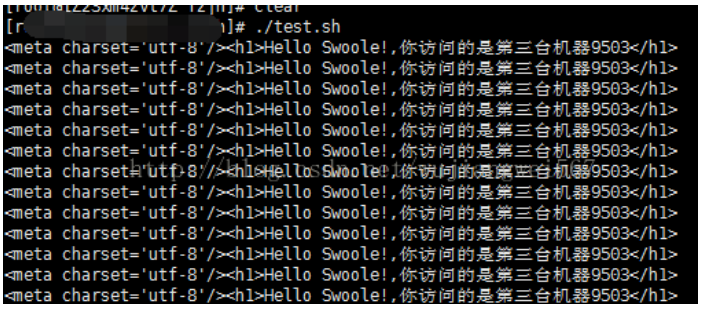
1. upstream backend {
2. server 127.0.0.1:9501 weight=65;
3. server 127.0.0.1:9502 weight=105;
4. server 127.0.0.1:9503 weight=45;
5. server 127.0.0.1:9504 weight=85;
6. ip\_hash;
7. }

只需要在最后一行加入ip\_hash;即可

我们不断刷新浏览器，始终访问到的是 服务器四



在终端中使用 curl 始终访问的是服务器 三



验证成功。