**operation 运维 day 02**

==================================

**静态网站 在不同环境下，浏览网站的内容不会发生变化**

**动态网站 在不同环境下，浏览网站的内容有可能会发生变化，该种类网站显示内容效果更好，会让用户体验更好，更适合目前市场环境**

**----------------------------------------------------**

**部署LNMP环境，有了该环境，nginx就可以很便利的支持动态网站，这也是主流的企业网站平台之一**

**L：Linux操作系统**

**N：Nginx网站服务软件**

**M：MySQL、MariaDB数据库**

**P：网站开发语言**

**LNMP（Linux、Nginx、MySQL、PHP）**

**部署LNMP环境**

**1，准备nginx**

rm -rf /usr/local/nginx/ //删除原有环境

netstat -ntulp | grep :80 //查询80端口的占用情况

killall nginx //如果nginx还在运行，可以杀掉

cd lnmp\_soft/nginx-1.17.6/

./configure --help | grep http\_ssl //查询模块名称

./configure --with-http\_ssl\_module //配置

make //编译

make install //安装

**2，安装其他相关软件包**

yum -y install mariadb mariadb-server mariadb-devel //安装

数据库的客户端、服务端、依赖包

yum -y install php //安装php解释器程序

yum -y install php-mysql //安装php与数据关联的软件包

yum -y install php-fpm //安装让nginx具有动态网站解析能

力的软件包

systemctl restart mariadb //开启数据库

systemctl restart php-fpm //开启php-fpm

netstat -ntulp | grep :3306 //查看数据库端口

netstat -ntulp | grep :9000 //查看php-fpm端口

**3，测试效果，目前还无法让nginx解析动态网站**

cd /root/lnmp\_soft/php\_scripts

cat test.php //查看文件，有普通的网站语言(html)，也有php语言

cp test.php /usr/local/nginx/html/ //拷贝动态网站页面到nginx的html目录下，使用火狐浏览器访问192.168.2.5/test.php可以看到下载文件的提示

**4，分析没有看到动态网站的原因：Nginx默认只支持静态页面**

用户如果访问静态页面，nginx会直接将该页面传递给用户的主机，用户再使用浏览器解析该页面内容

用户如果访问动态页面（如php语言编写的test.php页面），nginx要先将该页面扔给后台的php-fpm,该程序就会利用php解释器翻译php语言编写的网站页面，然后发给nginx程序，nginx再转发给用户，最终用户才能看到动态页面

cat /etc/redhat-release

CentOS Linux release 7.5.1804 (Core)

**5，nginx实现动静分离**

**为了让nginx支持动态网站的解析，要先修改配置文件**

第65到71行去掉注释，69行不用去，才可以实现动态页面

解析

65 location ~ \.php$ { //~是使用正则表达式，匹配以.php结尾

66 root html;

67 fastcgi\_pass 127.0.0.1:9000; //一旦用户访问了.php结尾的文件，就让nginx找后台的php-fpm（端口号9000）

68 fastcgi\_index index.php;

69 # fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi

\_script\_name;

70 include fastcgi.conf; //这里需要修改名称

71 }

-----------------------------

[root@proxy nginx]# sbin/nginx -s reload //重新加载nginx配置文件

再使用火狐浏览器访问192.168.2.5/test.php可以看具体页面内容了

**6，再测试调用数据库的网页**

首先拷贝页面

cd ~/lnmp\_soft/php\_scripts/

cp mysql.php /usr/local/nginx/html/

再用火狐访问<http://192.168.2.5/mysql.php>

**测试数据库内容发生变化之后网页是否变化**

mysql 登录数据库，手工创建dc账户

MariaDB [(none)]> create user dc@localhost identified by '123';

再用刷新http://192.168.2.5/mysql.php页面，会看到dc账户

----------------------------------------------------

**进一步了解php-fpm服务：**

**php-fpm 是fastCGI的进程管理器**

**fastCGI 快速公共网关接口，可以用来关联网站服务与解释器**

**了解php相关配置文件，目前无需配置**

vim /etc/php-fpm.d/www.conf //php有关的配置文件

listen = 127.0.0.1:9000 //此处配置决定了php-fpm服务针对什么ip与什么端口

pm.start\_servers = 5 //一上来开启的进程数量，pstree可以查看php-fpm的进程数量，如果修改了需要重启php-fpm服务

pm.max\_children = 50 //开启的fastCGI进程最大数量

-------------------------------------------------------

**地址重写：**

**可以实现在用户访问网站页面时推送其他页面的内容，或者跳转域名**

比如用户访问a.html页面 看到的是b.html页面

或者访问[www.abcd1234.com](http://www.abcd1234.com) 看到的是 [www.abc.com](http://www.abc.com)

**地址重写语法格式：**

**rewrite regex replacement flag**

**rewrite 旧地址(支持正则) 新地址 [选项]**

----------------------------------------------------

**准备素材**

[root@proxy nginx]# echo "nginx\_web\_a~" > html/a.html

[root@proxy nginx]# echo "nginx\_web\_b~" > html/b.html

**恢复配置文件为默认状态**

[root@proxy nginx]# cd conf/

[root@proxy conf]# cp nginx.conf.default nginx.conf

cp：是否覆盖"nginx.conf"？ y

**地址重写测试1： 相同网站内的页面跳转，地址栏不变**

打开nginx主配置文件，在第42行添加

rewrite ^/a.html$ /b.html; 当访问a页面时可以看到b页面的内容,此处的a.html匹配可以使用正则没有添加^和$之前，可以匹配/xyz/a.html 或 /a.htmlxyz添加了正则符号之后匹配的更精准，只针对/a.html进行重写。

sbin/nginx -s reload 重加载配置文件

使用火狐浏览器访问192.168.2.5/a.html 但看到了b.html的内容

**地址重写测试2： 相同网站内的页面跳转，地址栏发生变化**

打开nginx主配置文件，在第42行添加

rewrite /a.html$ /b.html redirect; //添加了redirect（重定向）选项

sbin/nginx -s reload 重加载配置文件

使用火狐浏览器访问192.168.2.5/a.html，可以看到页面换成b.html的同时，浏览器地址栏也发生了变化

**地址重写测试3：不同网站的地址跳转**

打开nginx主配置文件，在第42行添加

rewrite / http://www.tmooc.cn; 访问老网站，跳转到新的

sbin/nginx -s reload 重加载配置文件

使用火狐浏览器访问192.168.2.5/a.html 会跳到tmooc.cn

**地址重写测试4：不同网站的相同页面的地址跳转**

打开nginx主配置文件，在第42行添加

rewrite ^/(.\*)$ http://www.tmooc.cn/$1; 访问老网站的某个页面时，跳转到新网站对应的相同页面。前面使用正则表达式匹配用户输入的任意页面，并保存起来（小括号在正则中的效果是保留，相当于保存复制），后面使用$1将之前保存的页面地址粘贴到新网站

sbin/nginx -s reload 重加载配置文件

使用火狐浏览器访问192.168.2.5/a.html

**地址重写测试5：根据用户的情况决定跳转到什么样的页面**

cd /usr/local/nginx/html

mkdir firefox //创建火狐浏览器专属文件的目录

echo "proxy\_nginx\_firefox~" > firefox/test.html //准备火狐专用页面

cd ..

echo "proxy\_nginx\_other~" > html/test.html //准备普通浏览器使用的页面

vim conf/nginx.conf //删除原有地址重写的配置，然后在第46行添加

if ($http\_user\_agent ~\* firefox) { //如果用户的浏览器使用了火狐，就执行下面的rewrite任务，~代表匹配正则，\*是不区分大小写，$http\_user\_agent是nginx的内置变量，存储了用户的信息，比如用的什么浏览器

rewrite ^/(.\*)$ /firefox/$1; //就跳转到火狐专用目录的页面

}

sbin/nginx -s reload 重加载配置文件

分别使用火狐浏览器与其他浏览器访问192.168.2.5/test.html，可以得到两个不同页面内容则成功