operation 02

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

**一， 部署LNMP环境，实现动态网站解析**

**静态网站 在不同环境下，网站内容不会变化**

**动态网站 在不同环境下，网站内容有可能发生变化**

**LNMP 环境**

**L linux 操作系统**

**N nginx 网站服务**

**M mariadb(mysql) 数据库**

**P php 编写动态网站的语言工具**

**1，准备nginx以及相关软件包**

killall nginx //停止nginx程序

rm -rf /usr/local/nginx //删除nginx原有目录

cd /root/lnmp\_soft/nginx-1.17.6

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

make //编译

make install //安装

yum -y install mariadb mariadb-server //安装数据库客户端

与服务端

systemctl start mariadb //开启数据库服务

yum -y install mariadb-devel //安装数据库开发环境依赖包

yum -y install php //安装php，相当于解释器

yum -y install php-fpm //安装可以帮助nginx解析php语言编写的动

态网站的服务

yum -y install php-mysql //安装php与mysql关联的软件包

systemctl start php-fpm //开启php-fpm服务

yum -y install net-tools

netstat -ntulp | grep mysql //检查数据库

netstat -ntulp | grep php-fpm //检查php-fpm服务

**用户 – 浏览器html语言 ------------------- 服务器 html页面**

**php 页面 --- php-fpm**

**2，准备动态网站页面的测试文件**

cp ~/lnmp\_soft/php\_scripts/test.php /usr/local/nginx/html //拷贝动态网站测试页面到nginx中

cd /usr/local/nginx

sbin/nginx //启动nginx服务

**打开nginx配置文件，第65到71行去掉注释（可以用:65,71s/#//），69行不用去**

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

root html; //网站页面位置，不用改，保持默认

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）

fastcgi\_index index.php; //动态网站的默认页面，无需修改

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

//无用行，保持注释状态

include fastcgi.conf; //这里是另外一个配置文件，需要改扩展名

}

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

**fastCGI 快速公共网关接口，可以连接如nginx等网站程序到网站**

**的语言解释器(比如php)**

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

sbin/nginx //开启nginx，如果已经开启就使

用sbin/nginx -s reload 如果均不能正常开启，就用killall nginx然后重新试

使用火狐访问<http://192.168.2.5/test.php> 可以看到页面内容

**测试有数据库的动态网站**

cp ~/lnmp\_soft/php\_scripts/mysql.php /usr/local/nginx/html/ //拷贝另外一个测试页面到nginx

浏览器打开http://192.168.2.5/mysql.php 可以看到网站显示了数据的账户信息

mysql //进入数据库

create user dc@localhost identified by '123'; //创建测试账户

quit //退出

刷新http://192.168.2.5/mysql.php 可以看到新创建的用户

**看到以上test.php、mysql.php两个页面说明nginx不但能识别静态网站，也能解析动态**

**网站了，这种情况也可以记作nginx实现了动静分离**

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

**二，地址重写**

**可以定义用户的访问路径可以看到的实际内容**

**rewrite 匹配路径 实际看到的页面 选项**

**地址重写测试1：相同网站不同页面**

cp conf/nginx.conf.default conf/nginx.conf //可以先还原配置文件

cp：是否覆盖"conf/nginx.conf"？ y

打开配置文件，在38行添加

rewrite /a.html /b.html; //用户访问的路径中包含a.html的话

就跳转到b.html页面

然后准备测试页面

echo "nginx-a~~" > html/a.html

echo "nginx-b~~" > html/b.html

[root@proxy nginx]# sbin/nginx -s reload

使用浏览器访问192.168.2.5/a.html看到的是b.html的内容

**地址重写测试2：相同网站不同页面**

rewrite ^/a.html$ /b.html redirect; //在刚刚的配置中添加redirect

sbin/nginx -s reload

使用http://192.168.2.5/a.html路径访问网站时，地址栏同时发生

变化

**地址重写测试3：不同网站间跳转**

rewrite / http://www.tmooc.cn; //访问192.168.2.5的网站可以

跳转到www.tmooc.cn

sbin/nginx -s reload

**地址重写测试4：不同网站间跳转**

rewrite /(.\*) http://www.tmooc.cn/$1; //访问老网站会跳到新

网站，同时会携带所访问的页面

sbin/nginx -s reload

**地址重写测试5：不同浏览器跳转到不同页面**

火狐专用页面 192.168.2.5/abc.html html/firefox/abc.html

其他专用页面 192.168.2.5/abc.html html/abc.html

修改配置文件，删除原有地址重写，原地添加

if ($http\_user\_agent ~\* firefox){ //如果用户使用了火狐浏览器

rewrite /(.\*) /firefox/$1; //就进行地址重写操作，让用户看到火狐专属页面

}

//$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

~匹配正则 \*忽略大小写

改完后sbin/nginx -s reload

使用火狐浏览器查看192.168.2.5/abc.html可以看到之前html/firefox目录下的页

面，非火狐浏览器打开192.168.2.5/abc.html看到的是html下的页面

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

**选项**

**redirect 临时重定向 状态码 302 爬虫不感兴趣**

**permanent 永久重定向 状态码 301 爬虫感兴趣**

以下两种写法对于用户来说效果一样，但是对百度这样的网站不同

rewrite ^/a.html$ /b.html permanent;

rewrite ^/a.html$ /b.html redirect;

**last 不再读其他rewrite**

echo "nginx-c~~" > html/c.html //准备素材c页面

rewrite /a.html /b.html last; //不加last的话nginx会把多个rewrite语句综合处理

结果就是看a页面结果跳到c页面了，加了last的话可以避免这个情况

rewrite /b.html /c.html ;

**break 不再读其他语句**

location / { //此处为默认的location

rewrite /a.html /b.html break; //将last改为break可以阻止后面的语句，此处

如果是last则无法阻止后面location语句中的rewrite语句

root html;

index index.html index.htm;

}

location /b.html { //这里是新添加的location

rewrite /b.html /c.html;

}