|  |
| --- |
| 文章标题 |
| **作者：**  **归档：学习笔记** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文 |
| **格式说明：**  蓝色字体：注释  黄色背景：重要  绿色背景：注意 |

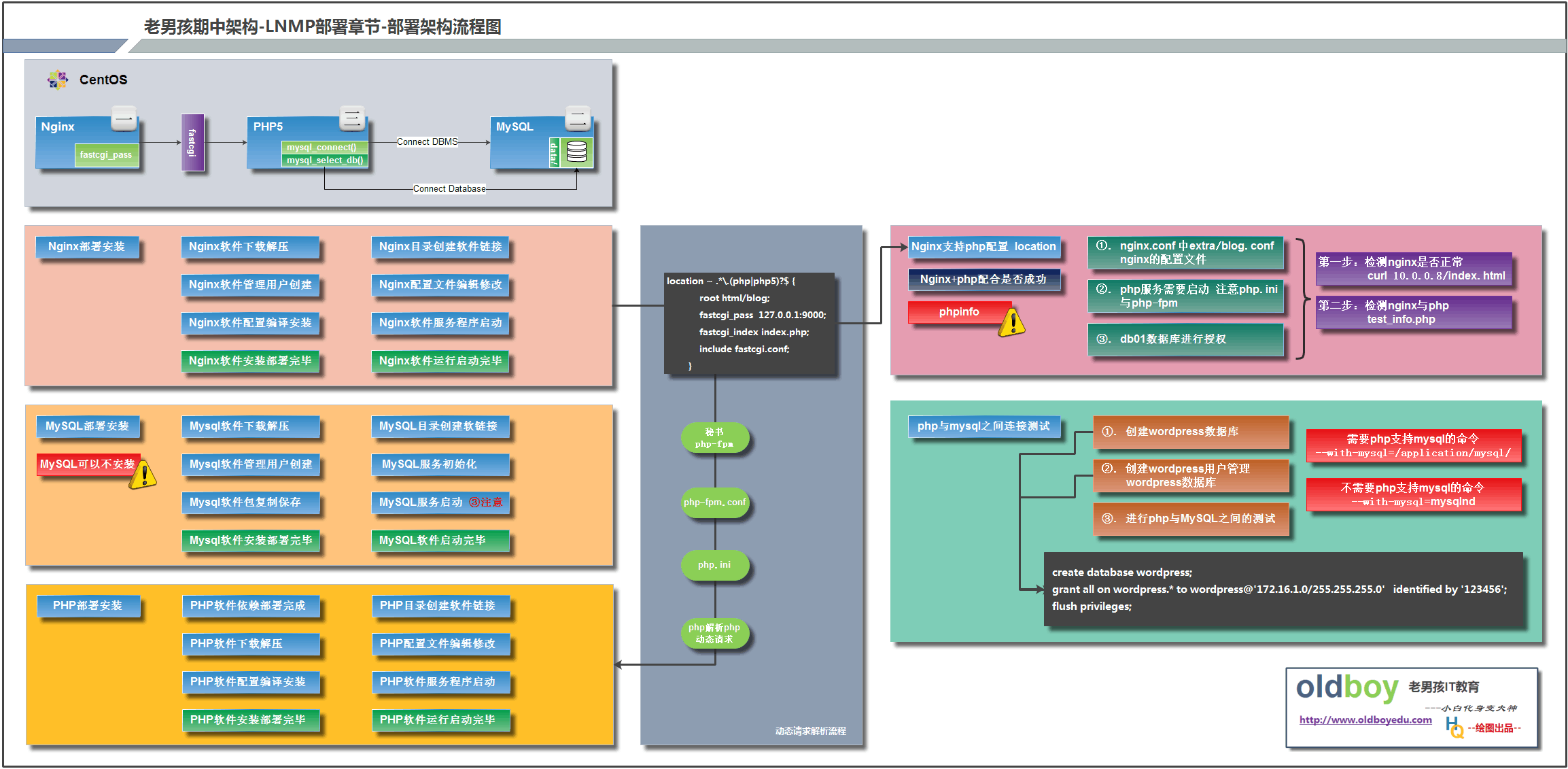
# 期中架构LNMP架构部署

L: linux

N: nginx

M: MySQL

p: PHP



## LNMP FastCGI知识说明

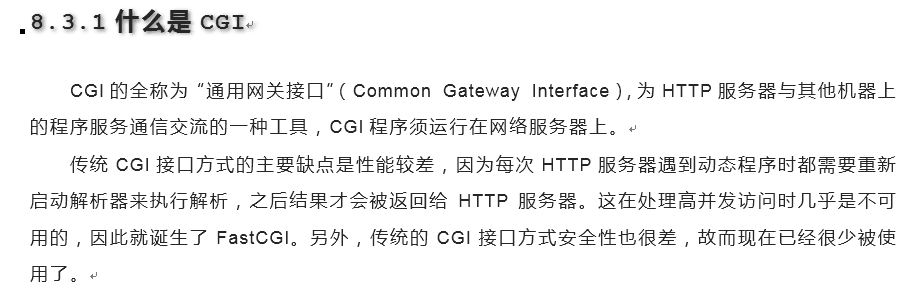
### 工作原理讲解说明:

①. 用户请求的静态文件，由nginx服务自行处理，根据静态的location配置进行处理

用户请求的动态文件，由php服务进行处理，根据动态的location配置进行处理

②. nginx服务接收到动态请求，会将请求抛送给fastcgi，类似于nginx服务接收动态请求的秘书，秘书会将动态请求送给PHP程序

③. PHP如果可以处理，会将处理结果直接通过fastcgi返回给nginx程序；如果不可以处理，还会请求后端数据库，最终再把处理结果返回给nginx



## LNMP MySQL数据库安装（二进制包的安装）

#### mysql软件安装方式：

01. yum安装mysql

02. 编译安装mysql（期中架构后详细讲解）

03. 二进制包安装mysql

#### mysql软件下载地址信息

www.mysql.com

mirrors.sohu.com

### mysql 安装步骤

#### 下载解压mysql二进制软件包

cd /home/oldboy/tools

tar xf mysql-5.6.34-linux-glibc2.5-x86\_64.tar.gz

#### 添加mysql程序运行时的管理用户

useradd -s /sbin/nologin -M mysql

#### 将mysql解压后的程序包搬家到程序目录下，并进行重命名

mkdir -p /application/

mv /server/tools/mysql-5.6.34-\*-x86\_64 /application/mysql-5.6.34

#### 为mysql服务程序创建软链接

ln -s /application/mysql-5.6.34/ /application/mysql

#### 让mysql用户管理 /application/mysql/

chown -R mysql.mysql /application/mysql/data

#### 初始化数据库

#1. 确认软件安装在哪里

#2. 确认数据存放在哪里

#3. 确认MySQL使用的用户谁？

/application/mysql/scripts/mysql\_install\_db --basedir=/application/mysql --datadir=/application/mysql/data --user=mysql

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

To start mysqld at boot time you have to copy

support-files/mysql.server to the right place for your system

mysql启动脚本 默认放在support-files/mysql.server

记得给MySQL设置个密码

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !

To do so, start the server, then issue the following commands:

创建密码

/application/mysql/bin/mysqladmin -u root password '填写密码'

指定用户修改

/application/mysql/bin/mysqladmin -u root -h web01 password 'new-password'

修改密码

格式：mysqladmin -u用户名 -p旧密码 password 新密码

#### 复制启动脚本 授权

cp /application/mysql/support-files/mysql.server /etc/init.d/mysqld

chmod +x /etc/init.d/mysqld

#### 修改启动脚本 和 mysql命令中的路径

sed -i 's#/usr/local/mysql#/application/mysql#g' /application/mysql/bin/mysqld\_safe /etc/init.d/mysqld

#### 复制默认的配置文件

\cp /application/mysql/support-files/my-default.cnf /etc/my.cnf

/etc/init.d/mysqld start

至此，mysql数据库安装与启动完成；可以检测进程和服务端口，确认mysql服务是否启动正确

### mysql 优化

#### 简化进入mysql数据库方法：

默认安装后进入命令

/application/mysql/bin/mysql

扩展知识：登录数据库命令简化方法

echo 'export PATH=/application/mysql/bin:$PATH' >>/etc/profile

source /etc/profile

which mysql

退出数据库时，尽量不要用ctrl+c进行退出mysql 用ctrl+d进行退出

exit

#### 设置mysql服务开机自启动

chkconfig --add mysqld

chkconfig mysqld on

#### 设置mysql数据库登录用户名与密码

记得给MySQL设置个密码

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !

To do so, start the server, then issue the following commands:

创建密码

/application/mysql/bin/mysqladmin -u root password '填写密码'

指定用户修改

/application/mysql/bin/mysqladmin -u root -h web01 password 'new-password'

修改密码

格式：mysqladmin -u用户名 -p旧密码 password 新密码

给MySQL root用户设置密码

/application/mysql/bin/mysqladmin -u root password 'oldboy123'

mysql -uroot -poldboy123

#### 数据库基础操作（数据库框架）

show databases; <--- 查询默认的数据库信息

use mysql; <--- 表示选择使用一个数据库，相当于cd进入一个数据库

select database(); <--- 表示查看当前所在数据库，类似于pwd命令的功能

select user(); <--- 查看当前登录数据库的用户，类似于whoami命令

并且mysql还可以限制指定用户可以从哪里进行连接登录数据库

select user,host from mysql.user; ---查看可以登录mysql数据库的目录，以及都可以从哪里进行管理mysql数据库

初级 添加删除系列

#创建数据库

create database wordpress;

#删除数据库

drop database wordpress;

## LNMP PHP软件安装

### 解决PHP软件的依赖关系

yum install zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel -y

yum install freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel libxslt-devel -y

rpm -qa zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel

#### libiconv软件安装--和字符集转换相关的软件

#mkdir -p /server/tools

cd /server/tools

#wget http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.14.tar.gz

tar zxf libiconv-1.14.tar.gz

cd libiconv-1.14

./configure --prefix=/usr/local/libiconv

make

make install

cd ../

#### 和数据加密相关的三个软件安装

#wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo

yum -y install libmcrypt-devel mhash mcrypt

rpm -qa libmcrypt-devel mhash mcrypt

### 下载解压PHP软件

cd /server/tools/

tar xf php-5.5.32.tar.gz

cd php-5.5.32

./configure \

--prefix=/application/php-5.5.32 \

--with-mysql=/application/mysql/ \

--with-pdo-mysql=mysqlnd \

--with-iconv-dir=/usr/local/libiconv \

--with-freetype-dir \

--with-jpeg-dir \

--with-png-dir \

--with-zlib \

--with-libxml-dir=/usr \

--enable-xml \

--disable-rpath \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-inline-optimization \

--with-curl \

--enable-mbregex \

--enable-fpm \

--enable-mbstring \

--with-mcrypt \

--with-gd \

--enable-gd-native-ttf \

--with-openssl \

--with-mhash \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-soap \

--enable-short-tags \

--enable-static \

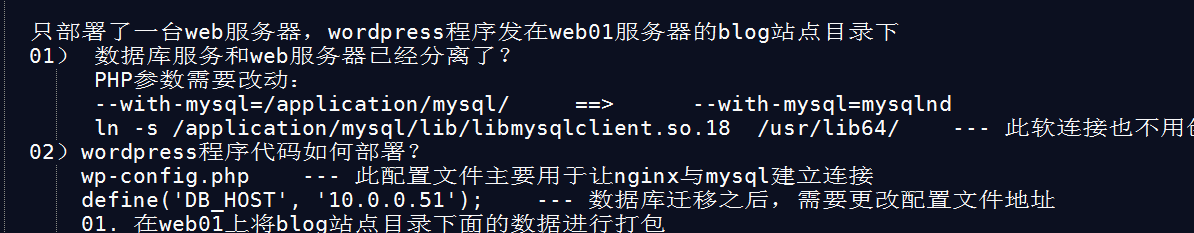
--with-xsl \

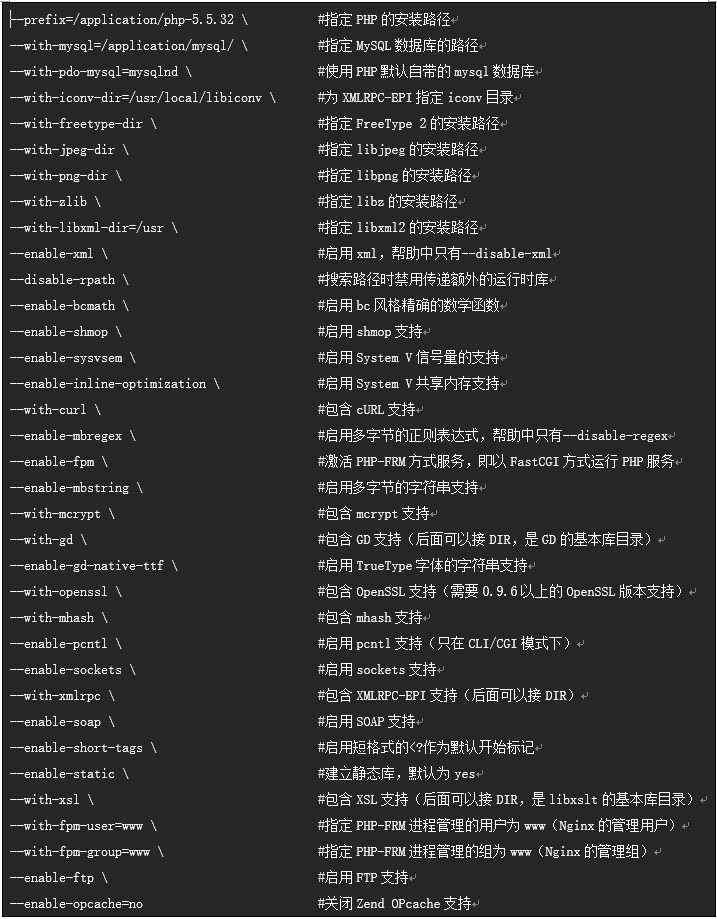
--with-fpm-user=www \

--with-fpm-group=www \

--enable-ftp \

--enable-opcache=no





#Generating files

#configure: creating ./config.status

#creating main/internal\_functions.c

#creating main/internal\_functions\_cli.c

#+--------------------------------------------------------------------+

#| License: |

#| This software is subject to the PHP License, available in this |

#| distribution in the file LICENSE. By continuing this installation |

#| process, you are bound by the terms of this license agreement. |

#| If you do not agree with the terms of this license, you must abort |

#| the installation process at this point. |

#+--------------------------------------------------------------------+

#

#Thank you for using PHP.

#### 防错

ln -s /application/mysql/lib/libmysqlclient.so.18 /usr/lib64/

touch ext/phar/phar.phar

#### 编译安装

make

make install

### PHP软件程序创建软链接

ln -s /application/php-5.5.32/ /application/php

### 配置php解析文件/配置php-fpm配置文件

cd /server/tools/php-5.5.32

cp php.ini-production /application/php/lib/php.ini

php.ini-production 与 php.ini-development 文件区别关系对比

扩展：文件比较命令 diff zdiff vimdiff windows上的文件比较

cd /application/php/etc/

cp php-fpm.conf.default php-fpm.conf

### 启动php-fpm程序

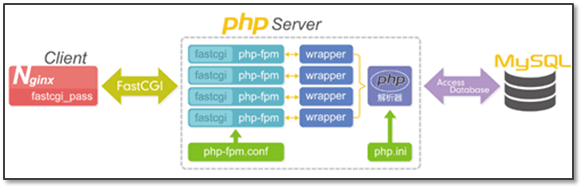
/application/php/sbin/php-fpm

确认php 9000端口是否正确启动

lsof -i :9000

ps -ef|grep php-fpm

## wordpress博客站点部署配置



### 修改nginx配置文件，使nginx程序与php程序建立联系

vim extra/blog.conf

server {

listen 80;

server\_name blog.etiantian.org;

location / {

root html/blog;

index index.php index.html index.htm; <-- 需要注意编辑修改默认首页文件 index

}

location ~\* .\*\.(php|php5)?$ {

root html/blog;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

}

说明：利用nginx的location区块实现动态请求与静态请求的分别处理

### 编辑nginx与php连通性测试文件,并进行测试

echo '<?php phpinfo(); ?>' >/application/nginx/html/blog/test\_info.php

#### 测试站点

curl http://blog.etiantian.org/index.html <-- 静态请求站点文件信息测试

curl http://blog.etiantian.org/test\_info.php <-- 动态请求站点文件信息测试

说明：当php服务停止时，9000端口信息消失，即停止PHP错误报502错误

linux系统测试完毕后，建议利用浏览器进行最终测试，测试效果更明显些

### 编辑php与mysql连通性测试文件,并进行测试

#### 创建数据库

mysql -uroot -poldboy123;

show databases; <--- 查看当前数据库信息

create database wordpress;

#### 添加用户信息

grant all on wordpress.\* to 'wordpress'@'localhost' indentified by 'oldboy123';

grant all on wordpress.\* to 'wordpress'@'172.16.1.%' indentified by 'oldboy123';

flush privileges;

drop user wordpress@'172.16.1.8'; <--- 删除用户信息

select user,host from mysql.user; <--- 查看用户信息

mysql -uwordpress -p123456 <--- 测试创建的用户连接

show databases; <--- 查看当前数据库信息

#### 测试php与数据库连通性

vim test\_mysql.php

<?php

//$link\_id=mysql\_connect('主机名','用户','密码');

//mysql -u用户 -p密码 -h 主机

$link\_id=mysql\_connect('localhost','wordpress','oldboy123') or mysql\_error();

if($link\_id){

echo "mysql successful by oldboy !\n";

}else{

echo mysql\_error();

}

?>

#### 测试php与数据库的连通性

curl blog.etiantian.org/test\_mysql.php

## 数据库迁移

### 数据库服务和nginx服务不在一台服务器上如何部署LNMP

01. mysql软件安装部署

02. 创建数据库wordpress和授权wordpress用户

03：将nginx服务器上的mysql数据库中的数据进行备份

mysqldump

04：将nginx服务器上数据库备份远程传输到mysql服务器上

05：利用MySQL命令恢复备份数据到本地数据库中

至此：实现数据库服务迁移

#### 备份数据库数据库信息

mysqldump -uroot -poldboy123 --all-databases >/tmp/bak.sql

ll /tmp/bak.sql -h

scp /tmp/bak.sql 172.16.1.51:/tmp/

#### 恢复数据库数据库信息

##db01

mysql -uroot -poldboy123 </tmp/bak.sql

###db01添加新的用户

grant all on wordpress.\* to wordpress@'172.16.1.0/255.255.255.0' identified by 'oldboy123';

flush privileges;

mysql -uwordpress -poldboy123 -h 172.16.1.51

#### 数据库迁移完毕，修改网站连接数据库的配置文件

mysql -uwordpress -poldboy123 -h 172.16.1.51 <-- 修改配置文件之前，先测试网站web服务器与迁移后的数据库连通性

vim wp-config.php <-- 修改wordpress上的数据库连接参数信息

/\*\* MySQL主机 \*/

define('DB\_HOST','172.16.1.51') <-- 修改连接的主机信息，将localhost修改为172.16.1.51

说明：web服务器数据库此时可以关闭了

### 如何将用户上传的数据保存到NFS服务器中

#### 在nginx服务器上找到用户上传的数据，存储在什么目录下了

upload

1）：网页上右键查看图片信息，图片链接信息里面就指定了upload目录

2）：find /application/nginx/html/blog -type f -mmin -1

02. 先把upload目录中的数据移到/tmp

03. 将upload目录作为一个挂载点，挂载到nfs指定共享目录上（用户映射的问题）

04. 在将/tmp/目录中用户上传数据移回到upload目录中

将用户上传的数据存储到了NFS共享服务器中了

至此：用户上传数据迁移过程完毕，所有数据都存储在了NFS共享服务器中

#### 只部署了一台web服务器，wordpress程序发在web01服务器的blog站点目录下

01） 数据库服务和web服务器已经分离了？

PHP参数需要改动：

--with-mysql=/application/mysql/ ==> --with-mysql=mysqlnd

ln -s /application/mysql/lib/libmysqlclient.so.18 /usr/lib64/ --- 此软连接也不用创建

02）wordpress程序代码如何部署？

wp-config.php --- 此配置文件主要用于让nginx与mysql建立连接

define('DB\_HOST', '10.0.0.51'); --- 数据库迁移之后，需要更改配置文件地址

01. 在web01上将blog站点目录下面的数据进行打包

02. 通过网络将打包数据分发到web02、web03服务器上

03. 在web02 web03上解压数据包，将数据直接放置到本地blog目录中

迁移数据库：利用数据库备份命令

#### 进行数据迁移到NFS共享服务器

01：先将原有目录中数据移出

cd /application/nginx/html/blog/wp-content/uploads

mkdir /tmp/wordpress\_backup -p

mv ./\* /tmp/wordpress\_backup/

02：NFS服务器上配置创建共享目录

vim /etc/exports

/data 172.16.1.0/24(rw,sync,all\_squash)

showmount -e 172.16.1.31

mount -t nfs 172.16.1.31:/data /mnt/

showmount -e 172.16.1.31

mount -t nfs 172.16.1.31:/data/ ./uploads/

mv /tmp/wordpress\_backup/\* ./