

# Linux高级运维

**NSD OPERATION**

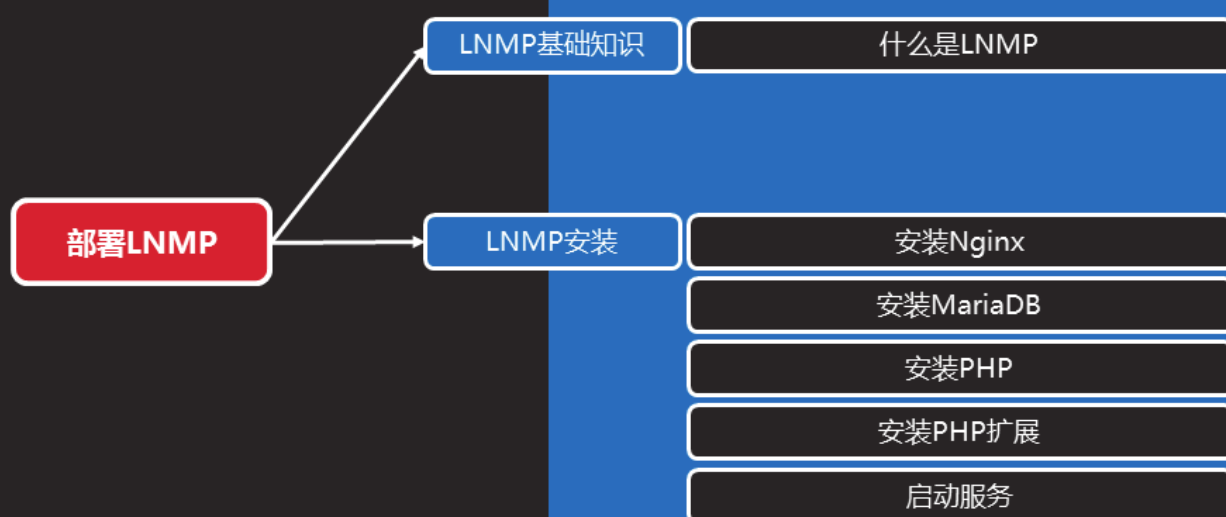
**DAY02**

# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	部署LNMP
	10:30 ~ 11:20	Nginx+FastCGI
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	Nginx高级技术
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



## 部署LNMP



# LNMP基础知识

## 什么是LNMP

- 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(PHP、Perl、Python)



# LNMP安装



## 安装Nginx

```
[root@localhost ~]# tar -zxvf nginx-1.12.2.tar.gz  
[root@localhost ~]# cd nginx-1.12.2
```

知识讲解

```
[root@localhost nginx-1.12.2]# ./configure \  
> --prefix=/usr/local/nginx \  
> --with-http_ssl_module \  
[root@localhost nginx-1.12.2]# make && make install
```



# 安装MariaDB

知识讲解

```
[root@localhost ~]# yum -y install mariadb-server  
[root@localhost ~]# yum -y install mariadb  
[root@localhost ~]# yum -y install mariadb-devel
```



# 安装PHP

知识讲解

```
[root@localhost ~]# yum -y install php  
[root@localhost ~]# yum -y localinstall php-fpm-5.4.16-42.el7.x86_64.rpm
```



# 安装PHP扩展

```
[root@localhost ~]# yum -y install php-mysql
```

//给PHP安装扩展库文件，使得PHP可以连接MariaDB数据库

知识讲解



# 启动服务

- 启动Nginx服务

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx
```

```
[root@localhost ~]# netstat -utnlp | grep :80
```

```
tcp      0      0 0.0.0.0:80  0.0.0.0:*    LISTEN  32428/nginx
```

知识讲解



## 启动服务（续1）

知识讲解

- 启动MySQL服务

```
[root@localhost ~]# systemctl start mariadb  
[root@localhost ~]# systemctl status mariadb  
[root@localhost ~]# netstat -utnlp | grep :3306
```

- 启动PHP服务

```
[root@localhost ~]# systemctl start php-fpm  
[root@localhost ~]# systemctl status php-fpm  
[root@localhost ~]# netstat -utnlp | grep :9000
```



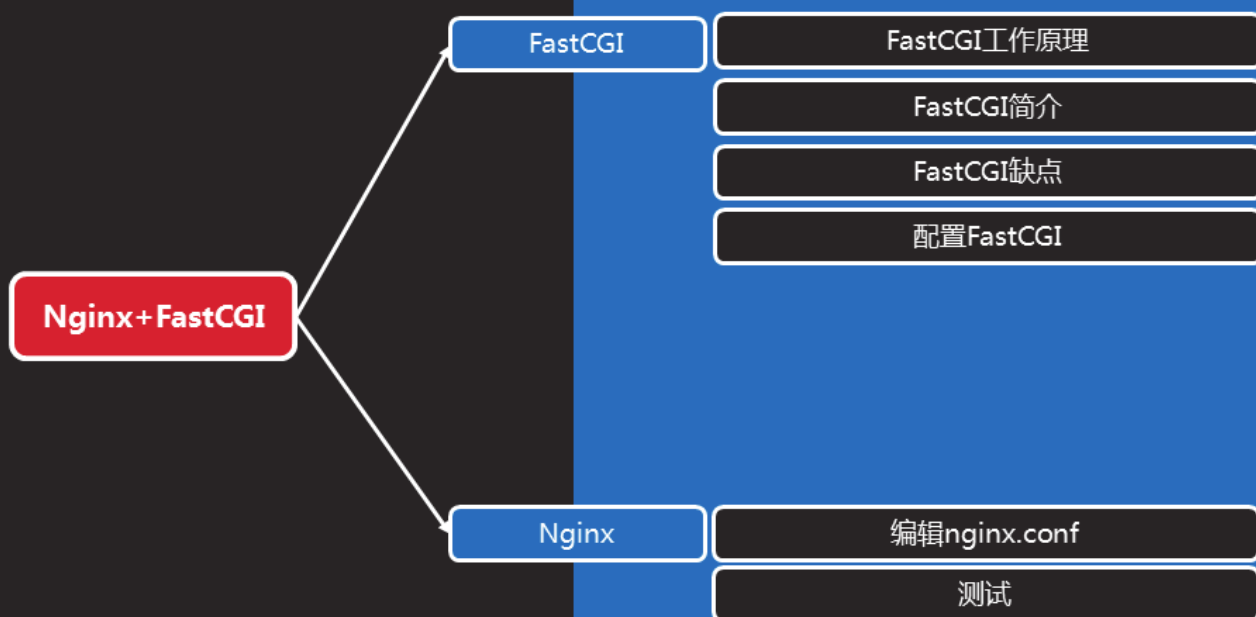
## 案例1：部署LNMP环境

课堂练习

- 任务要求
  - 安装Linux、Nginx、MySQL、PHP环境
  - 启动Nginx、MySQL、PHP服务
  - 测试Nginx及MySQL、PHP是否工作正常

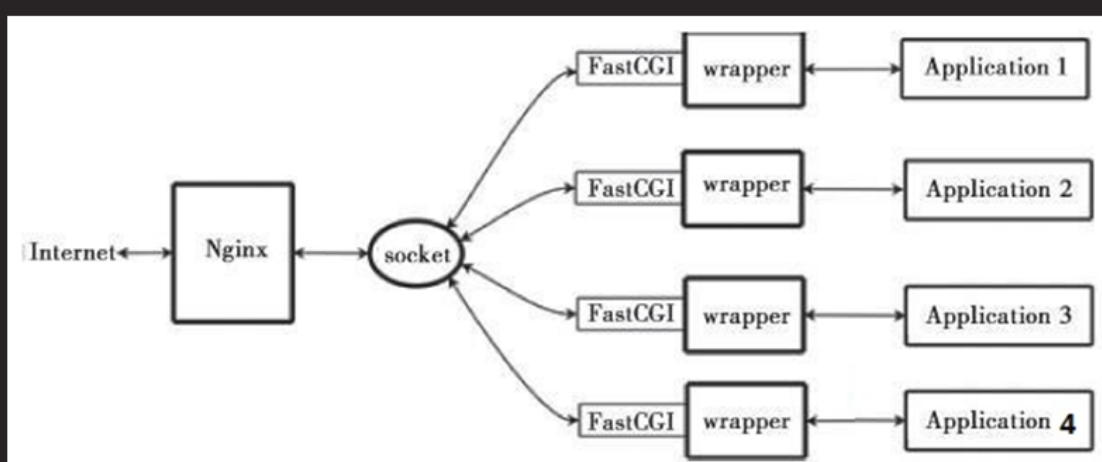


# Nginx+FastCGI



## FastCGI工作原理

Tedu.cn  
达内教育



知识讲解





## FastCGI工作原理（续1）

知识讲解

- 工作流程
  - 1. Web Server启动时载入FastCGI进程管理器
  - 2. FastCGI进程管理器初始化，启动多个解释器进程
  - 3. 当客户端请求到达Web Server时，FastCGI进程管理器选择并连接到一个解释器
  - 4. FastCGI子进程完成处理后返回结果，将标准输出和错误信息从同一连接返回Web Server



## FastCGI简介

知识讲解

- FastCGI技术目前支持语言有 PHP、C/C++、Java、Perl、Python、Ruby等



## FastCGI缺点

知识讲解

- 内存消耗大
  - 因为是多进程，所以比CGI多线程消耗更多的服务器内存，PHP-CGI解释器每进程消耗7至25兆内存，将这个数字乘以50或100就是很大的内存数
  - Nginx+PHP(FastCGI)服务器在3万并发连接下
    - 开10个Nginx进程消耗150M内存 ( 10\*15M )
    - 开64个php-cgi进程消耗1280M内存 ( 20M\*64 )



## 配置FastCGI

知识讲解

```
[root@localhost ~]# vim /etc/php-fpm.d/www.conf
[www]
listen = 127.0.0.1:9000
listen.allowed_clients = 127.0.0.1

user = apache
group = apache
pm = dynamic
pm.max_children = 50
pm.start_servers = 5
pm.min_spare_servers = 5
pm.max_spare_servers = 35
```



# Nginx

## 编辑nginx.conf

```
[root@localhost ~]# vim /usr/local/nginx/conf/nginx.conf
```

```
location / {
```

```
    root html;
```

```
    index index.php index.html index.htm;
```

```
}
```

```
location ~ \.php$ {
```

```
    root html;
```

```
    fastcgi_pass 127.0.0.1:9000;
```

//php-fpm的IP与端口

```
    fastcgi_index index.php;
```

```
    #fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```
    include fastcgi.conf;
```

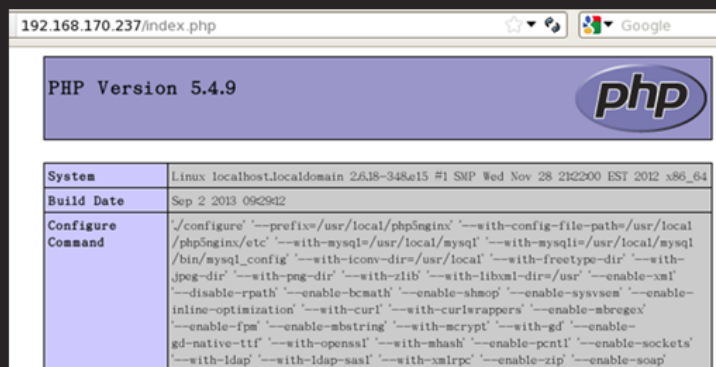
//加载Fast-CGI参数文件

```
}
```

# 测试

- 创建PHP首页index.php，测试效果

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx -s reload
[root@localhost ~]# systemctl restart php-fpm
[root@localhost html]# vim /usr/local/nginx/html/test.php
<?php
$i=33;
echo $i;
?>
```



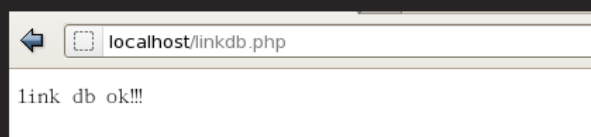
知识讲解



## 测试（续1）

- 创建PHP页面，测试数据库连接效果

```
[root@localhost html]# vim /usr/local/nginx/html/mysql.php
<?php
$mysqli = @new mysqli('localhost', 'root', '', 'mysql');
//主机名，用户名，密码，数据库
if ($mysqli->connect_errno) {
    die('Connect Error: ' . $mysqli->connect_errno);
}
else{
    echo "link db OK!!!!";
}
?>
```



知识讲解



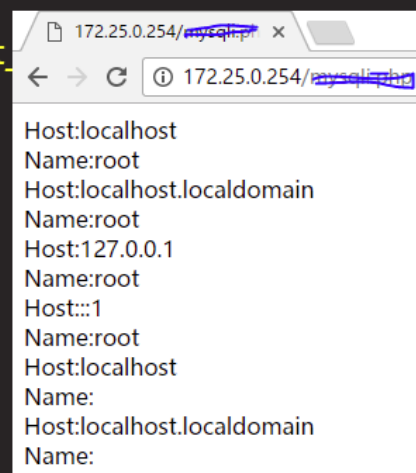
## 测试（续2）

- 创建PHP页面，测试数据库连接效果

```
[root@localhost html]# vim /usr/local/nginx/html/mysql.php
```

```
<?php
$mysqli = new mysqli('localhost','root','','mysql');
if (mysqli_connect_errno()){
    die('Unable to connect!'). mysqli_connect_
}
$sql = "select * from user";
$result = $mysqli->query($sql);
while($row = $result->fetch_array()){
    printf("Host:%s",$row[0]);
    printf("</br>");
    printf("Name:%s",$row[1]);
    printf("</br>");
}
?>
```

知识讲解



```
Host:localhost
Name:root
Host:localhost.localdomain
Name:root
Host:127.0.0.1
Name:root
Host:::1
Name:root
Host:localhost
Name:
Host:localhost.localdomain
Name:
```



## 案例2：构建LNMP平台

### 安装及部署LNMP平台

- 1) 配置Fast-CGI支持PHP网页
- 2) 创建PHP测试页面，测试使用PHP连接数据库的效果

课堂练习



# Nginx高级技术

Nginx高级技术

地址重写

基础知识

rewrite语法

应用案例

正则表达式

rewrite选项

**Tedu.cn**  
达内教育

## 地址重写

## 基础知识

知识讲解

- 什么是地址重写
  - 获得一个来访的URL请求，然后改写成服务器可以处理的另一个URL的过程
- 地址重写的好处
  - 缩短URL，隐藏实际路径提高安全性
  - 易于用户记忆和键入
  - 易于被搜索引擎收录



## rewrite语法

知识讲解

- rewrite基本语句
  - rewrite **regex** replacement **flag**
  - if ( 条件 ) {...}



## 应用案例

- 要求：a.html --> b.html

```
[root@localhost ~]# cat /usr/local/nginx/conf/nginx.conf
```

```
... ..
```

```
location / {  
    root html;  
    index index.html index.htm;  
    rewrite "/a.html$" /b.html;  
}
```

知识讲解



## 应用案例（续1）

- 要求：\*.jpg或\*.gif --> logo.png

```
[root@localhost ~]# cat /usr/local/nginx/conf/nginx.conf
```

```
... ..
```

```
location / {  
    root html;  
    index index.html index.htm;  
    rewrite \.(gif|jpg)$ /logo.png;  
}
```

知识讲解





## 应用案例（续2）

- 要求：域名跳转

– www.tarena.com --> bbs.tarena.com

知识讲解

```
[root@localhost ~]# cat /usr/local/nginx/conf/nginx.conf
server {
    listen    80;
    server_name www.tarena.com;
    location / {
        root html;
        index index.html index.htm;
        rewrite ^/(.*) http://bbs.tarena.com/$1;
    }
}
```



## 应用案例（续3）

- 要求：根据浏览器返回不同的页面

知识讲解

```
[root@localhost ~]# cat /usr/local/nginx/conf/nginx.conf
server {
    listen    80;
    server_name www.tarena.com;
    location / {
        root html;
        index index.html index.htm;
    }
    if ($http_user_agent ~ firefox) {
        rewrite ^/(.*)$ /nginx-firefox/$1 break;
    }
}
```



# 正则表达式

知识讲解

- 正则表达式匹配模式如下：
  - 区分大小写匹配：~
  - 不区分大小写匹配：~\*
  - 区分大小写不匹配：!~
  - 不区分大小写不匹配：!~\*



# rewrite选项

知识讲解

- rewrite语句
  - rewrite **regex** replacement **flag**
  - **flag** : **break**、**last**、**redirect**、**permanent**
  - last：停止执行其他重写规则，根据URI继续搜索其他 location，地址栏不改变
  - break：停止执行其他的重写规则，完成本次请求
  - redirect：302临时重定向，地址栏改变，爬虫不更新URI
  - permanent：301永久重定向，地址栏改变，爬虫更新URI



## 案例3：地址重写

课堂练习

- 配置重写策略，实现以下要求：
- 所有访问a.html的请求，重定向到b.html;
- 所有访问192.168.4.5的访问重定向至www.tmooc.cn；
- 所有访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面；
- 实现firefox与curl访问相同页面文件，返回不同的内容。



### 总结和答疑

总结和答疑

动态页面故障

问题现象

原因分析

# 动态页面故障

## 问题现象

- 故障错误信息

[root@svr5 nginx-1.8.0]# firefox http://192.168.2.100/test.php  
访问动态页面后，无法查看页面内容，浏览器显示下载窗口

[root@svr5 nginx-1.8.0]# firefox http://192.168.2.100/test.php  
访问动态页面后，显示空白页

# 原因分析

知识讲解

- 分析故障
  - 默认nginx无法解析PHP页面
  - nginx对自己无法识别的页面，则直接交给用户下载
  - 访问动态页面时，nginx返回空白页
- 分析故障原因
  - nginx没有配置fastcgi\_pass
  - 返回空白页时，请检查PHP页面文件的语法格式是否正确

