**老男孩linux运维实战培训**

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

**版权声明：**

本文作者为《老男孩linux运维实战培训》学生—**你的名字**

本文的所有内容均来自老男孩培训**命令总结**，未经本人及老男孩培训许可，禁止私自转发及使用。

QQ: **你的**XXXX

E-mail: **你的**XXXX

**联系方式:**

|  |  |  |  |
| --- | --- | --- | --- |
| 网站运维交流群： | | | |
| **Linux QQ交流群 385168604** | | **架构师 QQ交流群 390642196** | |
| **Python QQ交流群 29215534** | | **大数据 QQ交流群 421358633** | |
| =================================================================================== | | | |
| **老男孩linux实训联系方式** |  | |  |
| **咨询QQ** | **41117397（丹丹）** | | **70271111（歪歪）** |
|  | **80042789（飞雪）** | | **390320151（小雨）** |
|  | **41117483（冰冰）** | |  |
| **电话：** | **158-1059-0206（丹丹）** | | **189-1171-8229（歪歪）** |
|  | **135-5261-2571（飞雪）** | | **186-0046-2391（小雨）** |
| **网站:** | <http://www.etiantian.org> | | <http://www.oldboyedu.com> |
| **博客:** | http://oldboy.blog.51cto.com | | <http://blog.oldboyedu.com> |

**快捷方式说明:**

**ctrl + 1 一级标题**

**ctrl + 2 二级标题**

**ctrl + 3 三级标题**

**ctrl + 5 程序代码**

**ctrl + 6 正文**

**格式约定：**

蓝色字体：内容注释

目 录

[LNMP课程总结 1](#_Toc511393632)

[第1章 LNMP流行架构及原理流程 1](#_Toc511393633)

[1.1 LNMP组合工作原理 1](#_Toc511393634)

[第2章 Nginx介绍 2](#_Toc511393635)

[2.1 Nginx的重要特性和应用场合 2](#_Toc511393636)

[2.1.1 重要特性： 3](#_Toc511393637)

[2.1.2 Nginx企业级应用 3](#_Toc511393638)

[第3章 Nginx安装 6](#_Toc511393639)

[3.1 Nginx实战 6](#_Toc511393640)

[3.1.1 安装Nginx所需的pcre库 6](#_Toc511393641)

[3.1.2 Nginx配置文件解析： 9](#_Toc511393642)

[3.2 Nginx常用模块 13](#_Toc511393643)

[3.2.1 关于nginx模块 13](#_Toc511393644)

[3.2.2 nginx目录结构说明 14](#_Toc511393645)

[3.2.3 Nginx主配置文件 15](#_Toc511393646)

[3.2.4 Nginx虚拟机配置实战 16](#_Toc511393647)

[3.2.5 Nginx重启后的检测策略 20](#_Toc511393648)

[3.2.6 常用功能的配置 20](#_Toc511393649)

[3.2.7 配置nginx虚拟主机别名企业场景 21](#_Toc511393650)

[3.3 Nginx日志及错误日志详解 23](#_Toc511393651)

[3.3.1 日志切割：-写成脚本 24](#_Toc511393652)

[3.4 Nginx Location 26](#_Toc511393653)

[3.4.1 Location作用 26](#_Toc511393654)

[3.4.2 Location语法 26](#_Toc511393655)

[3.4.3 Nginx rewrite 27](#_Toc511393656)

[第4章 Mysql 数据库 29](#_Toc511393657)

[4.1 Mysql安装 29](#_Toc511393658)

[4.2 修改配置文件 30](#_Toc511393659)

[4.3 指定配置文件 30](#_Toc511393660)

[4.4 启动数据库 30](#_Toc511393661)

[4.5 启动关闭测试 30](#_Toc511393662)

[4.5.1 进入数据库 31](#_Toc511393663)

[第5章 FastCGI 34](#_Toc511393664)

[5.1.1 LNMP运行过程和解析原理 34](#_Toc511393665)

[第6章 PHP部署实战 35](#_Toc511393666)

[6.1 php安装 36](#_Toc511393667)

[6.1.1 php安装参数 36](#_Toc511393668)

[6.1.2 编译 37](#_Toc511393669)

[6.1.3 Nginx Mysql Php组合 39](#_Toc511393670)

[6.1.4 配置PHP连接数据库 41](#_Toc511393671)

[6.1.5 wordpress博客搭建实战 42](#_Toc511393672)

[6.1.6 实现wordpress博客URL静态化 42](#_Toc511393673)

[6.2 将web服务器数据库迁移到DB库 44](#_Toc511393674)

LNMP课程总结

集群架构图



集群脚本清单

|  |  |  |
| --- | --- | --- |
| 脚本名称 | 作用区域 | 备注 |
| 01\_one\_rsync\_Server.sh | Backup服务器部署配置 | /srv/salt/scripts |
| 02\_two\_rsync\_client.sh | Nfs服务器部署配置 | /srv/salt/scripts |
| 03\_Inotify+SersynCrsync.sh | Inotify+sersync数据同步配置 | /srv/salt/scripts |
| mysql\_db.sh | 数据库服务器部署 | /srv/salt/scripts |
| yum-one\_lnmp.sh | LNMP | /srv/salt/scripts |
| yum-one\_lamp.sh | LAMP | /srv/salt/scripts |
| nginx\_lb01.sh | 负载均衡node1 | /srv/salt/scripts |
| nginx\_lb02.sh | 负载均衡node2 | /srv/salt/scripts |
| yumRepository | YUM仓库部署 | /server/scripts |
| saltstack | Saltstack-master-minion部署 | \* |
| 04\_expect\_fenfa\_m01 | M01管理分发机器-YUM仓库 |  |

# LNMP流行架构及原理流程

LNMP（Linux Nginx MySQL PHP）近年新的web服务组合

LAMP (Linux Apache MySQL PHP) 比较早。比较经典的web服务组合

LEMP （Linux (Engine x) MYsql PHP)

## LNMP组合工作原理

* **当用户通过浏览器输入域名请求NginxWeb服务，如果请求的是静态数据。则Nginx直接解析返回给客户；动态数据请求（.php结尾），Nginx就会把它通过FastCGI接口发送给PHP引擎服务（FatstCGI进程php-fpm）进行解析，如果这个动态请求要读取数据库，则PHP就会继续向后请求Mysql数据库，以读取需要的数据，并最终通过Nginx服务把获取的数据返回给用户**

Nginx(-location(匹配 ~\*\.php${}))----fastCGI\_pass ip:9000----php





# Nginx介绍

Nginx、支持高性能、高并发的www服务和代理服务软件。工作在网络OSI第七层应用层；因为Nginx具有高并发（特别是静态资源）、占用系统资源少等特性，且功能丰富，在功能应用方面，Nginxbu不但是一个优秀的Web服务软件，还具有反向代理负载均衡功能和缓存服务功能。部署简单，方便；

## Nginx的重要特性和应用场合

Nginx基本特性：

* 针对静态资源高速高并发访问及缓存
* 可使用反向代理加速，可进行数据缓存
* 具有简单负载均衡、节点健康检查和容错
* 支持远程FastCGI的缓存加速
* 支持SSL、TLS、SNI
* 具有模块化的架构
* 具备WWW服务特性：
* 基于IP加端口的多虚拟主机站点
* 支持Keep-ailved
* 简单、方便、灵活的配置管理
* 自定义日志格式
* 支持rewrite模块，URL重写及正则表达式
* 支持基于客户端IP地址和HTTP基本认知的访问控制
* 支持HTTP响应限速
* 支持同一IP地址的并发连接或请求数限制
* 支持邮件服务代理

### 重要特性：

* 支持高并发：能支持几万并发连接
* 资源消耗少：在3万并发连接下，开启10个Nginx线程消耗的内存不到200MB
* 可以做HTTP反向代理及加速缓存，即负载均衡，内置对RS节点服务器健康检查功能
* 具备Squid等专业缓存软件的缓存功能
* 支持异步网络I/O事件模型epoll

### Nginx企业级应用

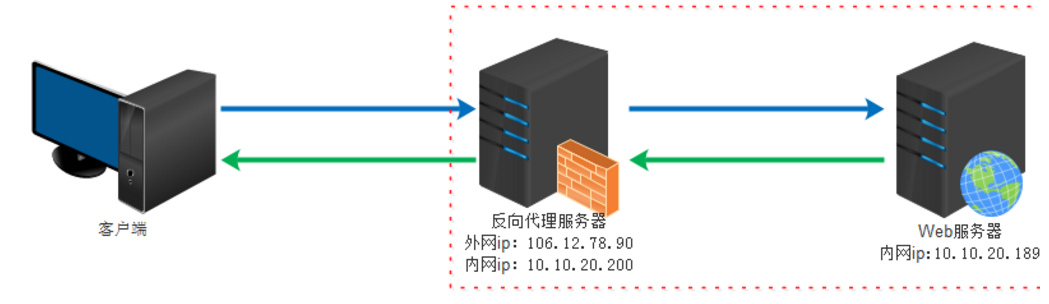
* 作为Web服务软件，支持更多的并发连接访问，占用资源更少，效率高
* 反向代理或负载均衡服务

反向代理：代替外网用户向集群内部请求服务

正想代理：代替内网用户访问外部业务

* 前端业务数据缓存服务

注释：反向代理的含义是：proxy代替客户向内部请求服务



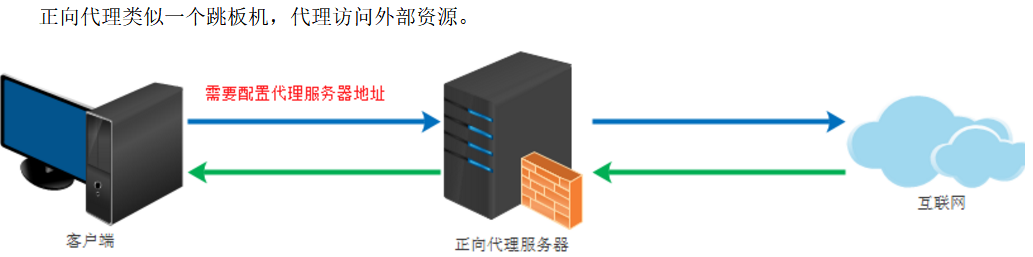
注释：正向代理:从局域网向外网请求服务，由代理（路由器）替局域网机器请求服务

正反向总结：

反向代理对用户来说是透明的，像是直接访问到服务器（其实不是直接访问的）

正向代理：用户通过proxy访问正常状态下访问不到的资源

　总结来说：正向代理 是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端必须要进行一些特别的设置才能使用正向代理。



* 缓存服务 SQUID,VARNISH



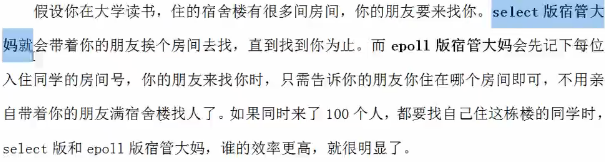
#### Nginx作为Web服务器主要应用场景：

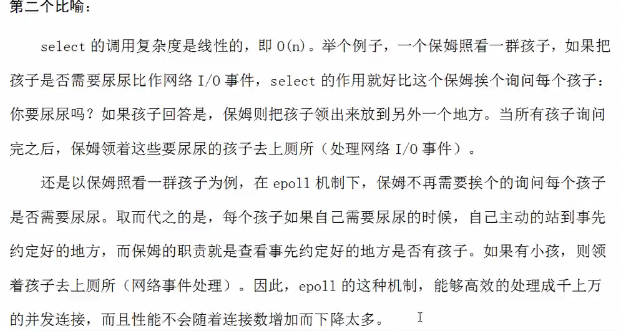
* 使用Nginx允许HTML,JS,CSS，小图片等静态数据
* Nginx结合FastCGI允许PHP等动态程序fastcgi\_pass方式
* Nginx结合Tomcat/Resin等支出Java动态程序常用proxy\_pass方式

#### Nginx相对应apache的对比

* 基于异步网络I/O模型(epoll、kqueue)
* 具备支出高性能，高斌发的特性并发连接科大数万
* 对小文件高并发之陈浩
* 进程占用系统资源比较低
* 支出web、反向proxy、cache、三大重点功能，且很优秀

**Epoll模型与Select模型的形象比喻**





Web服务器程序的选择

静态业务：高并发场景，尽量采用Nginx或lighttpd 首选Nginx

动态业务：理论采用Nginx或Apche

动静业务都有就采用Nginx

建议：

静态业务：高并发，采用nginx或lighttpd，根据自己的掌握程度或公司要求

动态业务：采用nginx和apache均可

既有静态和动态业务：nginx或apache 不要多选要单选

动态业务可以由前端代理（haproxy），根据页面元素的类型，向后转发相应的服务器进行处理。

如果并发不大，对apache很熟悉，采用apache也是可以的，papche2.4版本也很强大，并发连接数也有增加，见后的压力测试

提示：nginx做web（apache,lighttpd）、反向代理（haproxy,lvs nat）及缓存服务器（squid）也是非常不错的

思想：学习，工作都不要追求一步到位。满足需求的前提下，先用，然后逐步完善

最终建议：对外业务使用nginx,对内业务Apachenginx（）

## Nginx安装

Linux里面软件的安装方法：

1. Rpm -ivh PackupNeam.rpm(需要解决依赖问题)
2. YUM安装是解决RPM包安装依赖的问题，简单 。

优点：简单、易用、高效

缺点：不能定制

1. 编译（C语言源码-编译二进制 ./configure 配置，make编译 ，make install 安装）

优点：可定制

缺点：复制、效率低

1. 定制制作rpm包，搭建yum仓库：将定制的rpm包放到yum仓库，进行yum安装。

优点：结合了2点的优点和3的优点

缺点：复制

### Nginx实战

#### 安装Nginx所需的pcre库

Pcre中文“perl兼容正则表达式”使Ngixn支持具备URL重写功能的Rewrite模块。

**yum install pcre pcre-devel -y**

<http://nginx.org/2015.html> 下载源码包，此次使用1.6.3版本

[root@web01 tools]# pwd

/home/oldboy/tools

[root@web01 tools]# wget -q <http://nginx.org/download/nginx-1.6.3.tar.gz>

[root@web01 **yum install openssl-devel -y**  安装加密程序

[root@web01 nginx-1.6.3]# **useradd www -s /sbin/nologin -M**  创建nginx用户

[root@web01 tools]# ll

总用量 788

-rw-r--r-- 1 root root 805253 4月 8 2015 nginx-1.6.3.tar.gz

**[root@web01 tools]# tar xf nginx-1.6.3.tar.gz**

[root@web01 tools]# **cd nginx-1.6.3**

[root@web01 nginx-1.6.3]# ls

auto CHANGES CHANGES.ru conf configure contrib html LICENSE man README src

[root@web01 nginx-1.6.3]## ./configure --user=www --group=www --prefix=/application/nginx1.6.3 --with-http\_stub\_status\_module --with-http\_ssl\_module

**指定安装路径，状态模块，ssl模块**

**[root@web01 nginx-1.6.3]# make && make install**

[root@web01 nginx-1.6.3]# **ln -s /application/nginx-1.6.3/ /application/nginx** 创建软连接

[root@web01 nginx-1.6.3]# ll /application/

总用量 4

lrwxrwxrwx 1 root root 25 4月 9 21:57 nginx -> /application/nginx-1.6.3/

drwxr-xr-x 6 root root 4096 4月 9 21:57 nginx-1.6.3

[root@web01 nginx-1.6.3]# **/application/nginx/sbin/nginx** 启动nginx

[root@web01 nginx-1.6.3]# **lsof -i :80** 检查启动端口

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

nginx 10236 root 6u IPv4 60475 0t0 TCP \*:http (LISTEN)

nginx 10237 www 6u IPv4 60475 0t0 TCP \*:http (LISTEN)

[root@web01 nginx-1.6.3]# curl -I 10.0.0.8 本机测试

HTTP/1.1 200 OK

Server: nginx/1.6.3

Date: Mon, 09 Apr 2018 14:00:17 GMT

Content-Type: text/html

Content-Length: 612

Last-Modified: Mon, 09 Apr 2018 13:57:16 GMT

Connection: keep-alive

ETag: "5acb713c-264"

Accept-Ranges: bytes

[root@web01 nginx-1.6.3]# curl -I 172.16.1.8 本机测试

HTTP/1.1 200 OK

Server: nginx/1.6.3

Date: Mon, 09 Apr 2018 14:00:23 GMT

Content-Type: text/html

Content-Length: 612

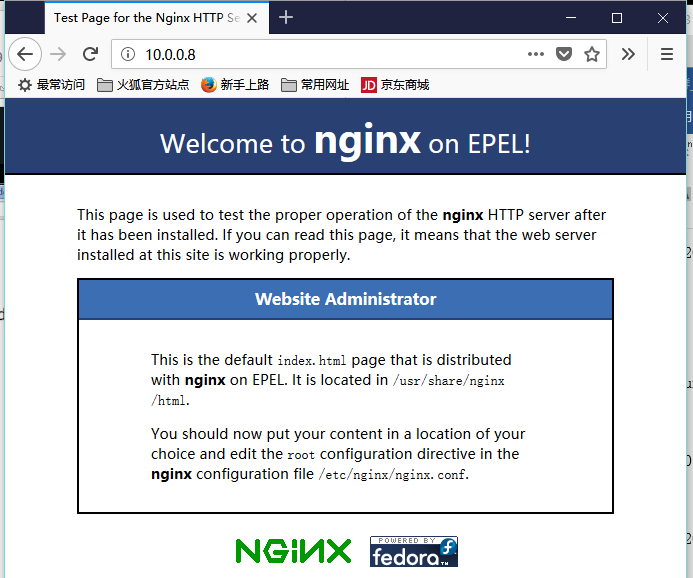
Last-Modified: Mon, 09 Apr 2018 13:57:16 GMT

Connection: keep-alive

ETag: "5acb713c-264"

Accept-Ranges: bytes

浏览器确认



Nginx安装常见错误记录

暂省略….

##### 排查三部曲：

* ping 192.168.33.130 物理通不通
* telnet 192.168.33.130 80 浏览器到web通不通
* 服务器本地curl 192.168.33.130 WEB服务开没开
* error.log 看错误日志

安装前执行如下操作

* 关闭selinux
* 关闭防火墙

#### Nginx配置文件解析：

/application/nginx/conf

[root@web01 conf]# vim nginx.conf

[root@web01 nginx]# cat conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

##### Html首页编写

[root@web01 html]# pwd

/application/nginx/html

[root@web01 html]# vim index.html

<html>

<meta charset="utf-8"/>

<head>

<title>老男孩教育28期-网络班-linux系统运维-程军</title>

</head>

<body >

第28期人物风云榜

<table border=1>

<tr>

<td>学号<td>

<td>姓名<td>

</tr>

<tr>

<td>007<td>

<td>程军<td>

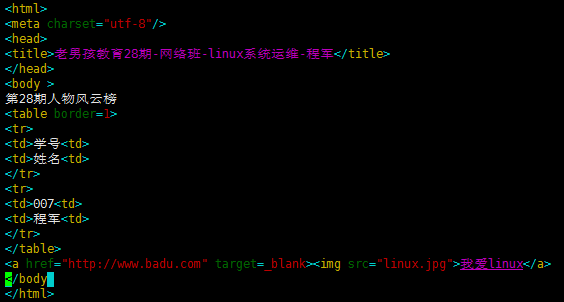
</tr>

</table>

<a href="http://www.badu.com" target=\_blank><img src="linux.jpg">我爱linux</a>

</body>

</html>



[root@web01 conf]# ***/application/nginx/sbin/nginx -t*** **检查语法**

nginx: the configuration file /application/nginx-1.6.3//conf/nginx.conf syntax is ok

nginx: configuration file /application/nginx-1.6.3//conf/nginx.conf test is successful

[root@web01 conf]# **curl 10.0.0.8 curl测试页面内容**

<html>

<head>

<title>老男孩教育28期-网络班-linux系统运维-程军</title>

</head>

<body >

第28期人物风云榜

<table border=1>

<tr>

<td>学号<td>

<td>姓名<td>

</tr>

<tr>

<td>007<td>

<td>程军<td>

</tr>

</table>

<a href="http://www.badu.com" target=\_blank><img src="linux.jpg">我爱linux</a>

</body>

</html>

[root@web01 conf]# **curl -I 10.0.0.8 检查响应状态码**

HTTP/1.1 200 OK

Server: nginx/1.6.3

Date: Tue, 10 Apr 2018 02:02:40 GMT

Content-Type: text/html

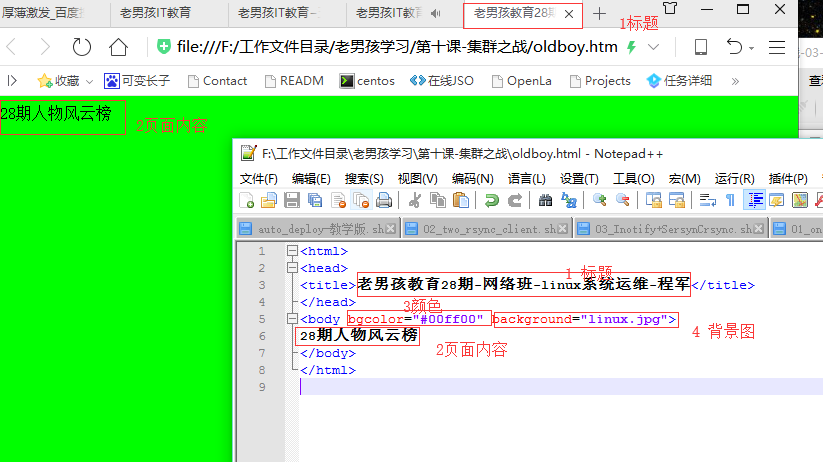
Content-Length: 369

Last-Modified: Tue, 10 Apr 2018 01:59:00 GMT

Connection: keep-alive

ETag: "5acc1a64-171"

Accept-Ranges: bytes



实际页面显示效果



局域网内部测试时注意在hosts文件中写映射

### Nginx常用模块

#### 关于nginx模块

Nginx使用不同的模块来实现不同的功能，主要有2组重要的模块

* Nginx core modules(必须的)

包括

Main

Events

Standard HTTP modules (虽然不是必需，但是缺省会安装，不建议改的)

典型包括

Core

Access

FastCGI

Gzip 压缩模块-性能优化

Log 日志模块

RewriteURL重写模块

Upstream 负载均衡

查看模块去看官方文档<http://nginx.org/en/docs>模块帮助



#### nginx目录结构说明

[root@oldboy nginx]# tree

.

├── client\_body\_temp

├── conf

│?? ├── **fastcgi.conf** <--**动态配置文件**

│?? ├── **fastcgi.conf.default**  <--**动态配置文件-备份**

│?? ├── fastcgi\_params <--**主参数文件**

│?? ├── fastcgi\_params.default

│?? ├── koi-utf

│?? ├── koi-win

│?? ├── mime.types

│?? ├── mime.types.default

│?? ├── nginx.conf <--**主配置文件 -静态**

│?? ├── nginx.conf.default <---**备份配置文件-静态**

│?? ├── scgi\_params

│?? ├── scgi\_params.default

│?? ├── uwsgi\_params

│?? ├── uwsgi\_params.default

│?? └── win-utf

├── fastcgi\_temp

├── html <--**默认的站点文件目录**

│?? ├── 50x.html

│?? └── index.html <--**默认的站点文件**

├── logs

│?? ├── access.log <--**访问日志**

│?? ├── error.log <--**错误日志**

│?? └── nginx.pid <--进程

├── proxy\_temp

├── **sbin**

│?? └── **nginx 启动 [-t 验证配置文件]**

├── scgi\_temp <-临时文件

└── uwsgi\_temp <-临时文件

9 directories, 21 files

#### Nginx主配置文件



[root@oldboy conf]# egrep -v "#|^$" nginx.conf

worker\_processes 1; 进程模式 数字指定cpu核数

events {

worker\_connections 1024; ngixn处理链接请求的最大数量

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80; 监控端口号

server\_name localhost; 域名/主机名

location / { 根据前边的结果执行下面的任务，执行RUL跳转

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html; 如遇错误是把错误提供给客户还是 /跳转到指定的页面

location = /50x.html { 如果为 /50x.htm页面侧从下面的 root html目录中找

root html;

}

} 使用最多的: 一个server标签就是一个虚拟主机

}

### Nginx虚拟机配置实战

#### 虚拟主机概念

在web服务里就是一个独立的网站站点，对应独立的域名也可能是IP或端口，具有独立的程序及资源目录，可以独立地对外提供服务供用户访问。

#### 虚拟主机类型

* 基于域名

以域名来区分 实例：[www.chengjun.com](http://www.chengjun.com)

* 基于端口

以不同的服务端口来区分不同的虚拟主机 实例：[www.chengjun.com:8080](http://www.chengjun.com:8080)

* 基于IP

通过不同的IP地址来区别虚拟主机

#### 基于域名虚拟主机实战：

[root@web01 conf]# cp nginx.conf nginx.conf.oldboy.20180410V1

[root@web01 conf]# cat nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.51cto.com;

location / {

root html/www;

index index.html index.htm;

}

}

server {

listen 80;

server\_name blog.51cto.com;

location / {

root html/blog;

index index.html index.htm;

}

}

}

[root@web01 conf]# mkdir ../html/{www,blog} -p

写入html文件

echo "blog" > ../html/blog/index.html

echo "www" > ../html/www/index.html

检查语法及配置

[root@web01 conf]# **/application/nginx/sbin/nginx -t 检查语法**

[root@web01 conf]# **/application/nginx/sbin/nginx -s reload 平滑重启**

页面测试



基于域名虚拟主机访问详细原理及过程

单机讲解：浏览器输入[www.51cto.com（默认80](http://www.51cto.com（默认80)端口）后，浏览器与Web服务先建立TCP连接（TCP三次握手）后，发送一个HTTP请求报文 ；Nginx 会响应，找www的虚拟主机。并向浏览器返回一个www资源的响应报文，浏览器进行显示

响应报文：

HTTP/1.1 200 OK

Server: nginx

Date: Tue, 10 Apr 2018 05:58:17 GMT

Content-Type: text/html; charset=UTF-8

Connection: keep-alive

Vary: Accept-Encoding

403问题，要么加首页，要么设置访问资源目录

访问及响应流程

**输入域名—>（发送TCP连接以及HTTP请求报文）解析IP及端口---🡪服务器Nginx响应请求报文读取请求头字段查找匹配配置文件中配置标签文件资源目录----Nginx返回响应报文【注意:没有请求头的话默认会匹配默认的第一条标签】**

#### 基于端口虚拟机

基于端口的虚拟机配置

server {

listen 81;

server\_name blog.51cto.com;

location / {

root html/blog;

index index.html index.htm;

}

}

server {

listen 82;

server\_name bbs.51cto.com;

location / {

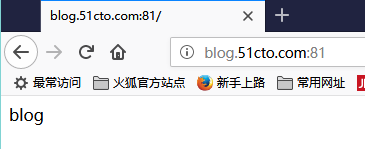
root html/bbs;

index index.html index.htm;

}

}

}



#### 基于IP虚拟机

设置子接口IP

[root@web01 conf]# ifconfig eth0:0 10.0.0.101/24 up 设置子接口IP

[root@web01 conf]# ip addr add 10.0.0.102/24 dev eth0 label eth0:1 设置别名IP

[root@web01 conf]# ifconfig eth0:0 down 关闭子接口IP

[root@web01 conf]# ip addr del 10.0.0.102/24 dev eth0 label eth0:1 删除子接口IP

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:0c:29:cc:03:ba brd ff:ff:ff:ff:ff:ff

inet 10.0.0.8/24 brd 10.0.0.255 scope global eth0

inet 10.0.0.101/24 brd 10.0.0.255 scope global secondary eth0:0

inet 10.0.0.102/24 scope global secondary eth0:1

inet6 fe80::20c:29ff:fecc:3ba/64 scope link

valid\_lft forever preferred\_lft forever

#### 基于IP虚拟机的nginx.conf配置文件

server {

listen 10.0.0.8:80;

server\_name www.51cto.com;

location / {

root html/www;

index index.html index.htm;

}

}

server {

listen 10.0.0.101:80;

server\_name blog.51cto.com;

location / {

root html/blog;

index index.html index.htm;

}

}

server {

listen 10.0.0.102:80;

server\_name bbs.51cto.com;

location / {

root html/bbs;

index index.html index.htm;

}

}

配置虚拟主机的流程：

* 复制一个完整的server标签段，到结尾 注意：要放在http的结束大括号前，也就是server标签段放入http标签
* 更改server\_name及对应网页的root根目录
* 检查配置文件语法，平滑重启服务
* 创建server\_name对应网页的根目录，并且建立测试文件，如果没有index首页会出现403错误
* 在客户端对应server\_name的主机名做host解析或DNS配置，并检查（ping域名看返回ip对不对）
* win浏览器访问，或者在linux客户端做host解析，用wget或curl访问

#### Nginx重启后的检测策略

#### 常用功能的配置

#### 配置文件优化

修改主配置文件如下

[root@web01 conf]# cat nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

include extra/\*.conf; 不使用正则匹配的话每次都需要修改主配置文件

include extra/index.conf;

include extra/www.conf;

include extra/blog.conf;

include extra/bbs.conf;

}

[root@web01 conf]# mkdir -p extra

[root@web01 conf]# sed -n '18,25p' nginx.conf.oldboy.baseportV2 >extra/www.conf

[root@web01 conf]# sed -n '26,33p' nginx.conf.oldboy.baseportV2 >extra/blog.conf

[root@web01 conf]# sed -n '34,41p' nginx.conf.oldboy.baseportV2 >extra/bbs.conf

[root@web01 conf]# tree extra/

extra/

├── bbs.conf

├── blog.conf

├── index.conf

└── www.conf

0 directories, 4 files

[root@web01 extra]# ll

总用量 12

-rw-r--r-- 1 root root 185 4月 10 15:33 bbs.conf

-rw-r--r-- 1 root root 187 4月 10 15:32 blog.conf

-rw-r--r-- 1 root root 179 4月 10 15:32 www.conf

### 配置nginx虚拟主机别名企业场景

nginx别名设置：通过空格的方式

server\_name [www.51cto.com](http://www.51cto.com) 51cto.com;

apache别名设置：

ServerAlias

#### nginx301重定向方法

worker\_processes 2;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name 51cto.com;

root html/www;

index index.html index.htm;

rewrite ^/(.\*) <http://www.51cto.com/$1> permanent; 通过别名永久跳转到该URL

}

server {

listen 80;

server\_name blog.51cto.com;

root html/blog;

index index.html index.htm;

}

#### Ngixn状态监控实战

[root@web01 extra]# cat status.conf

##satatus

server {

listen 80;

server\_name status.51cto.com;

location / {

stub\_status on;

access\_log off;

}

}

[root@oldboy conf]# ../sbin/nginx -t

[root@oldboy conf]# ../sbin/nginx -s reload

**访问status.51cto.com**



Active connections: 2 /nginx正处理的活动连接数

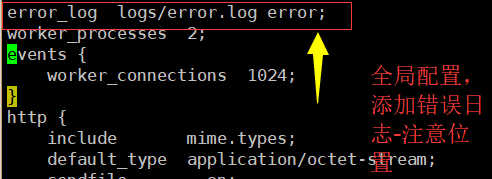
server accepts handled requests

65 65 55

Reading: 0 Writing: 1 Waiting: 3

* 第一个server表示nginx启动到现在共处理了的连接数
* 第二个accepts表示nginx启动到现在工成功创建的握手数次
* 请求丢失数=（握手数-连接数）如下表示没有丢失请求
* 第三个handled requests 表示总共处理了的请求连接数
* Reading:nginx读取到客户端的Header 信息数
* Writing:nginx 返回给客户端的Header 信息数

### Nginx日志及错误日志详解



error\_log logs/error.log error; <---全局配置日志 及日志等级

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" ' 添加日志配置

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

sendfile on;

keepalive\_timeout 65;

include extra/index.conf;

include extra/status.conf;

include extra/www.conf;

include extra/blog.conf;

include extra/bbs.conf;

}

单独配置日志示例；www.conf

server {

listen 80;

server\_name www.51cto.com 51cto.com ; <-域名 及 别名

root html/www; <---root用户，html存放路径

index index.html index.htm;

access\_log logs/www\_access.log; <-日志参数- 存放位置- 日志名称

}

日志调参，参考官方文档<http://nginx.org/en/docs/http/ngx_http_log_module.html>

日志格式：

* $remote\_addr 直接客户端地址
* $http\_x\_forwarded\_for间接客户端地址
* $remote\_user 远程客户端用户名
* $time\_local 记录访问时间及时区
* $request 返回状态，200,404,304等
* $body\_bytes\_sebts 发送的body字节数
* $http\_referer 引用页-从那个链接访问来的
* $http\_user\_agen客户端浏览器
* $ request 用户的请求，使用HTTP协议

#### 日志切割：-写成脚本

mkdir /server/scripts

cd /server/scripts

vim cut\_nginx\_log.sh

#!/bin/bash

DR="/application/nginx/logs/"

IP=172.16.1.41

cd $DR &&\

mv www\_access.log www\_access$(date +%F -d -1day).log

mv blog\_access.log blog\_access$(date +%F -d -1day).log

mv bbs\_access.log bbs\_access$(date +%F -d -1day).log

mv index\_access.log index\_access$(date +%F -d -1day).log

/application/nginx/sbin/nginx -s reload

##rsync to backup server

rsync -az $DR --delete rsync\_backup@$IP::backup/ --password-file=/etc/rsync.passwd

#del date before 7 day ago

find $DR -type f -name "\*.log" -time +7 |xargs rm -f

加入定时任务：

# crontab -e

##### cut nginx access log by test at 20171231 ########每天零点进行切割##############

00 00 \* \* \* /bin/sh /server/scripts/cut\_nginx\_log.sh >/dev/null 2>&1

#### 日志分析：

分析工具软件Awstats

网页测试方式：查看状态码

[root@asid-java ~]# curl -i -s -w %{http\_code} 192.168.11.66|head -10

HTTP/1.1 200 OK

Date: Fri, 26 Jan 2018 02:15:22 GMT

Server: Apache

X-Powered-By: PHP/7.0.26

Link: <http://192.168.11.66/wp-json/>; rel="https://api.w.org/"

X-TEC-API-VERSION: v1

X-TEC-API-ROOT: http://192.168.11.66/wp-json/tribe/events/v1/

X-TEC-API-ORIGIN: http://192.168.11.66

Set-Cookie: wfvt\_3327314756=5a6a8f3ae3956; expires=Fri, 26-Jan-2018 02:45:22 GMT; Max-Age=1800; path=/; HttpOnly

X-Frame-Options: SAMEORIGIN

* -I 仅测试HTTP头
* -m 10 最多查询10s
* -o /dev/null 屏蔽原有输出信息
* -s silent 模式，不输出任何东西
* -w %{http\_code} 控制额外输出

### Nginx Location

#### Location作用

根据用户请求的URL来执行不同的应用。就是根据用户请求的网站地址进行匹配

#### Location语法

Lication [ = | ~ | ~\*| ^~ |@ ] url {

……

}

URL可以是普通字符串地址路径，或者是正则表达式，匹配成功则执行后面大括号里的相关指令

|  |  |  |  |
| --- | --- | --- | --- |
| Location | [ = | ~ | ~\* | ^~ | @ ] | url | {…..} |
| 指令 | 匹配标识 | 匹配的网站网址 | 匹配URL后要执行的配置段 |

特殊字符记熟

|  |  |
| --- | --- |
| **~** | 区分大小写 |
| **~\*** | 不区分大小写 |
| **！** | 取反 |
| **^~** | 作用是在进行常规的字符串匹配检查后，不做正则表达式检查 |
| **@** | 内部调用 |

Lication配置实战

[root@web01 extra]# cat www.conf

server {

listen 80;

server\_name www.51cto.com;

location / {

return 401;

}

location = / { 匹配“=”号优先No1

return 402;

}

location /documents/ {

return 403;

}

location ^~/images/ {

return 404;

}

location ~\*\.(gif|jpg|jpeg)$ {

return 500;

}

access\_log logs/www.access.log main;

}

实战结论

[root@web01 extra]# curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.51cto.com

402 **“=”号匹配优先级最高**

[root@web01 extra]# curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.51cto.com

401

[root@web01 extra]# curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.51cto.com/index.html

401

[root@web01 extra]# curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.51cto.com/documents/documents.html

403

[root@web01 extra]# curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.51cto.com/documents/1.gif

500

**不用URL及特殊字符组合匹配优先级顺序**

|  |  |  |
| --- | --- | --- |
| 优先级 | 不用URL急特殊字符组合匹配顺序 | 匹配说明 |
| No1 | location = / { | 精确匹配 |
| No2 | location ^~/images/ { | 匹配常用 字符，不做正则匹配检查 |
| No3 | location ~\*\.(gif|jpg|jpeg)$ { | 正则匹配 |
| No4 | location /documents/ { | 匹配常规字符串，如果有正则优先匹配正则 |
| No5 | location / { | 所有locaton都不能匹配后的默认匹配 |

#### Nginx rewrite

是实现URL地址重写，需要PCRE软件的支持，即通过Perl兼容正则表达式语法进行规制匹配

Rewrite语法

指令语法： rewrite regex repkacement[flag];

默认值：none

应用位置：server location if

简单rewrite语句

rewrite ^/(.\*) <http://www.51cto.com/$1> permanent;

[root@web01 extra]# cat www.conf

server {

listen 80;

server\_name www.51cto.com;

location / {

rewrite ^/(.\*) http://www.baidu.com/$1 permanent;当访问51cto时自动永久跳到百度

}

access\_log logs/www.access.log main;

}

|  |  |
| --- | --- |
| Flag | 说明 |
| Last | 本条规制匹配完成后，继续向下匹配新的locationURL规制 |
| Break | 本条规制匹配完成即终止，不再匹配后面规制 |
| Redirect | 返回302临时重定向，浏览器会显示跳转后的URL |
| Permanent | 返回301永久跳转，浏览器会显示跳转后的URL |

Last和break用来实现URL重写，浏览器地址栏URL地址不变，但在服务器端访问的程序及路径发现变化。

Redirect和Permanent用来实现URL跳转，浏览器地址栏会显示跳转后的URL地址。

注意一次访问，做了跳转一次后相当于并发两次，也就是两次访问【PV】

### Rewrite企业应用场景

实战：

[root@web01 extra]# vim www.conf

server {

listen 80;

server\_name www.51cto.com 51cto.com;

location / {

root html/www;

index index.html index.html;

}

rewrite ^(.\*)/bbs/ http://bbs.51cto.com break;

access\_log logs/www\_access.log main gzip buffer=32k flush=5s;

}

结论：别名访问，后添加bbs字符串实现301跳转

Nginx 访问认证

省略……

# Mysql 数据库

是一种关系型数据库管理软件，关系型数据库特点是将数据库保存在不同的二维表（Excel）中，并且将这些表放入不同的数据库中，而不是把所有数据统一放在一个大仓库里，这样的设计增加了Mysql的读取速度，灵活性和可管理性也得到了很大提高。访问及管理Mysql数据库的最常用标准化语言为SQL结构化查询语言。

## Mysql安装

[root@web01 ~]# useradd -s /sbin/noligin mysql -M

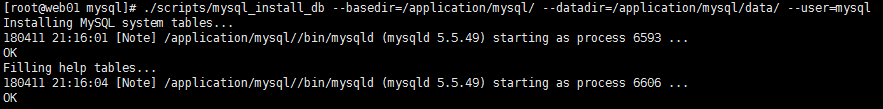
[root@web01 tools]# mv mysql-5.5.49-linux2.6-x86\_64 /application/mysql-5.5.49

[root@web01 tools]# cd /application/mysql-5.5.49/

[root@web01 tools]# ln -s /application/mysql-5.5.49/ /application/mysql

[root@web01 mysql]# chown -R mysql.mysql /application/mysql/

[root@web01 mysql]# ./scripts/mysql\_install\_db --basedir=/application/mysql/ --datadir=/application/mysql/data/ --user=mysql 🡨初始化



## 修改配置文件

cd /application/mysql/bin

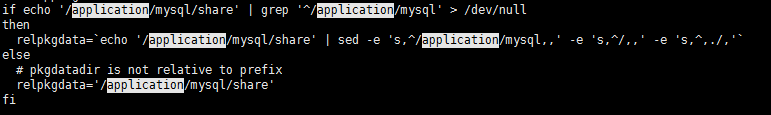
sed -i 's#/usr/local/#/application/#g' mysqld\_safe

sed -i 's#/usr/local/#/application/#g' support-files/mysql.server

cp support-files/mysql.server /etc/init.d/mysqld 拷贝到启动脚本

chmod +x /etc/init.d/mysqld 赋予权限

[root@web01 bin]# less mysqld\_safe



## 指定配置文件

[root@web01 mysql]# cp support-files/my-small.cnf /etc/my.cnf

cp：是否覆盖"/etc/my.cnf"？ y

## 启动数据库

[root@web01 mysql]# /application/mysql/bin/mysqld\_safe --user=mysql &

[1] 6953

180411 21:35:31 mysqld\_safe Logging to '/application/mysql/data/web01.err'.

180411 21:35:31 mysqld\_safe Starting mysqld daemon with databases from /application/mysql/data

PATH="/application/mysql/bin:$PATH" 设置环境变量

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

. /etc/profile

[root@web01 mysql]# lsof -i :3306

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

mysqld 7183 mysql 10u IPv4 61272 0t0 TCP \*:mysql (LISTEN)

## 启动关闭测试

[root@web01 mysql]# /etc/init.d/mysqld stop

Shutting down MySQL.180411 21:44:45 mysqld\_safe mysqld from pid file /application/mysql/data/web01.pid ended

SUCCESS!

[1]+ Done /application/mysql/bin/mysqld\_safe --user=mysql

[root@web01 mysql]# lsof -i :3306

[root@web01 mysql]# /etc/init.d/mysqld start

Starting MySQL... SUCCESS!

[root@web01 mysql]# lsof -i :3306

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

mysqld 7555 mysql 10u IPv4 62854 0t0 TCP \*:mysql (LISTEN)

[root@web01 mysql]# chkconfig mysqld on

### 进入数据库

[root@web01 mysql]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.5.49 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

[root@web01 ~]#

[root@web01 ~]# mysql -uroot -poldboy123

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 3

Server version: 5.5.49 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

安全起见删除历史脉络记录

History -d \*\*\*

#### Mysql数据库安装总结-脚化

useradd -s /sbin/noligin mysql -M

mkdir –p /application/

mkdir -p /home/oldboy/tools

cd /home/oldboy/tools/

tar zxf mysql-5.5.49-linux2.6-x86\_64.tar.gz

useradd -s /sbin/noligin mysql -M

mv mysql-5.5.49-linux2.6-x86\_64 /application/mysql-5.5.49

cd /application/mysql-5.5.49/

ln -s /application/mysql-5.5.49/ /application/mysql

chown -R mysql.mysql /application/mysql/

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

cd /application/mysql/bin

sed -i 's#/usr/local/#/application/#g' mysqld\_safe

cd ..

sed -i 's#/usr/local/#/application/#g' support-files/mysql.server

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

chmod +x /etc/init.d/mysqld

\cp support-files/my-small.cnf /etc/my.cnf

/application/mysql/bin/mysqld\_safe --user=mysql &

PATH="/application/mysql/bin:$PATH"

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

. /etc/profile

lsof -i :3306

/etc/init.d/mysqld stop

chkconfig mysqld on

#### 进入数据库

【mysql命令】

mysqladmin -u root password 'oldboy123' 设置数据库密码

mysql -uroot -poldboy123 进入数据库

mysqladmin -u root -poldboy123 password 123456 修改密码

create database wordpress;

show databases;

system whoami 查看用户

select user,host from mysql.user; 查看用户

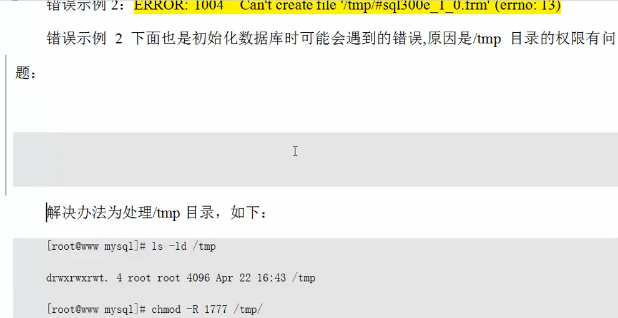
grant all on wordpress.\* to wordpress@'localhost' identified by '123456'; 创建管理用户

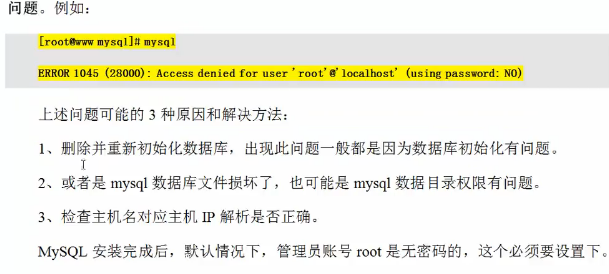
show grants for wordpress@'localhost'; 查看用户的权限

flush privileges; 刷新

#### mysql常见故障

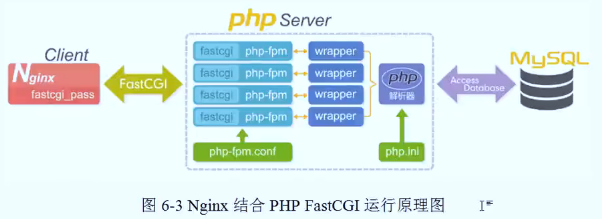
注意两个故障点1./tmp权限1777 2.主机名需要做解析



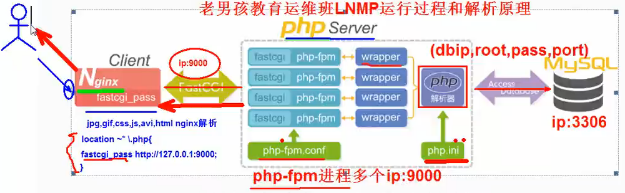


# FastCGI

快速通用网关接口(socket)，为HTTP服务器与其他机器上的程序服务通信交流工具常规web程序都支持。



### LNMP运行过程和解析原理



用户访问NginxWeb服务器-Ngixn负责解析静态数据（jpg,gif,css,js,avi,html）也接受动态数据(.php)动态数据通过FastCGI接口后抛（fastcgi\_pass监听 http://x.x.x.x:9000）给PHP服务器(php-fpm多个进程监听9000端口)[进程配置文件php-fpm.conf设置进程数量]监听到请求就会调用wrapper程序，wrapper会再启多个线程调用PHP解析器（引擎）进行解析，如要连数据库则向后数据库（dbip，root.pass.port3306）进行请求.数据库再将数据通过FastCGI将数据发给Nginx服务器后，Nginx再给用户

# PHP部署实战

1、确定nginx与数据库是启动的

[root@web01 ~]# netstat -tunlp|egrep "3306|80"

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 1657/mysqld

tcp 0 0 0.0.0.0:80

2、检查php所需的；lib库

[root@web01 ~]# rpm -qa zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel

libjpeg-turbo-devel-1.2.1-3.el6\_5.x86\_64

libpng-devel-1.2.49-2.el6\_7.x86\_64

freetype-devel-2.3.11-17.el6.x86\_64

gd-devel-2.0.35-11.el6.x86\_64

libcurl-devel-7.19.7-53.el6\_9.x86\_64

zlib-devel-1.2.3-29.el6.x86\_64

libxml2-devel-2.7.6-21.el6\_8.1.x86\_64

3、跟新yum源

更新bash源

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

更新epel源

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

4、yum安装lib库及哈希程序

yum install -y zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel libmcrypt-devel

###########

yum install -y mcrypt mhash

cd /home/oldboy/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

Centos7 下需要先操作如下

cd srclib/

sed -i -e '/gets is a security/d' ./stdio.in.h

|

make &&make install

5、再次确认基础库和扩展库程序是否安装齐全

[root@web01 tools]# rpm -qa zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel libmcrypt-devel mcrypt mhash

libxslt-devel-1.1.26-2.el6\_3.1.x86\_64

mhash-0.9.9.9-3.el6.x86\_64

libjpeg-turbo-devel-1.2.1-3.el6\_5.x86\_64

libmcrypt-devel-2.5.8-9.el6.x86\_64

libpng-devel-1.2.49-2.el6\_7.x86\_64

freetype-devel-2.3.11-17.el6.x86\_64

gd-devel-2.0.35-11.el6.x86\_64

libcurl-devel-7.19.7-53.el6\_9.x86\_64

zlib-devel-1.2.3-29.el6.x86\_64

libxml2-devel-2.7.6-21.el6\_8.1.x86\_64

mcrypt-2.6.8-10.el6.x86\_64

要一步一检查

## php安装

<http://mirrors.sohu.com/php/>

### php安装参数

useradd -s /sbin/noligin -M www

wget http://mirrors.sohu.com/php/php-5.5.32.tar.gz

tar zxf php-5.5.32.tar.gz

cd php-5.5.32

./configure \

--prefix=/application/php5.5.32 \ 安装路径

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

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

--with-freetype-dir \

--with-jpeg-dir \

--with-png-dir \

--with-zlib \

--with-libxml-dir=/usr \

--enable-xml \

--disable-rpath \

--enable-safe-mode \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-inline-optimization \

--with-curl \

--with-curlwrappers \

--enable-mbregex \

--enable-fpm \

--enable-mbstring \

--with-mcrypt \

--with-gd \

--enable-gd-native-ttf \

--with-openssl \

--with-mahsh \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-zip \

--enable-soap \

--enable-short-tags \

--enable-zend-multibyte \

--enable-static \

--with-xsl \

--with-fpm-user=nginx \ 用户

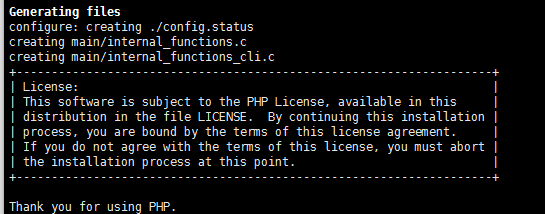
--with-fpm-group=nginx \ 组

--enable-ftp \ 开启ftp

--enable-opcache=no

./configure \

--prefix=/application/php5.5.32 \--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 \--with-openssl \--with-mhash \--enable-gd-native-ttf \--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



[root@web01 php-5.5.32]# ln -s /application/mysql/lib/libmysqlclient.so.18 /usr/lib64/

[root@web01 php-5.5.32]# touch ext/phar/phar.phar

### 编译

[root@web01 php-5.5.32]#make && make install





[root@web01 php-5.5.32]# ln -s /application/php5.5.32/ /application/php

[root@web01 php-5.5.32]# ls /application/php

bin etc include lib php sbin var

#### 设置php-fpm主配置文件

[root@web01 php-5.5.32]# cp php.ini-production /application/php/lib/php.ini

[root@web01 php5.5.32]# cd /application/php/etc/

[root@web01 etc]# cp php-fpm.conf.default php-fpm.conf

#### 启动php

[root@web01 ~]# /application/php/sbin/php-fpm

#### 检查是否启动

[root@web01 ~]# lsof -i :9000

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

php-fpm 30476 root 7u IPv4 155336 0t0 TCP localhost:cslistener (LISTEN)

php-fpm 30477 www 0u IPv4 155336 0t0 TCP localhost:cslistener (LISTEN)

php-fpm 30478 www 0u IPv4 155336 0t0 TCP localhost:cslistener (LISTEN)

[root@web01 ~]# netstat -tunlp|grep "php-fpm"

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN 30476/php-fpm

#### php安装总结

tar zxf php-5.5.32.tar.gz

cd php-5.5.32/

./configure \

--prefix=/application/php5.5.32 \

--with-mysql=/application/mysql \

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

--with-freetype-dir \

--with-jpeg-dir \

--with-png-dir \

--with-zlib \

--with-libxml-dir=/usr \

--enable-xml \

--disable-rpath \

--enable-safe-mode \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-inline-optimization \

--with-curl \

--with-curlwrappers \

--enable-mbregex \

--enable-fpm \

--enable-mbstring \

--with-mcrypt \

--with-gd \

--enable-gd-native-ttf \

--with-openssl \

--with-mahsh \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-zip \

--enable-soap \

--enable-short-tags \

--enable-zend-multibyte \

--enable-static \

--with-xsl \

--with-fpm-user=nginx \

--with-fpm-group=nginx \

--enable-ftp \

--enable-opcache=no

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

touch ext/phar/phar.phar

make && make install

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

cd /application/php/etc/

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

/application/php/sbin/php-fpm

### Nginx Mysql Php组合

cd /application/nginx/conf/extra

[root@web01 extra]# cat blog.conf

server {

listen 80;

server\_name blog.51cto.com;

location / {

root html/blog;

index index.html index.htm;

access\_log logs/blog\_access.log;

}

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

root html/blog;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

}

[root@web01 extra]# cd ../../html/blog/

[root@web01 blog]# echo "<?php phpinfo(); ?>" >test\_info.php

[root@web01 blog]# tree

.

├── index.html

└── test\_info.php

0 directories, 2 files

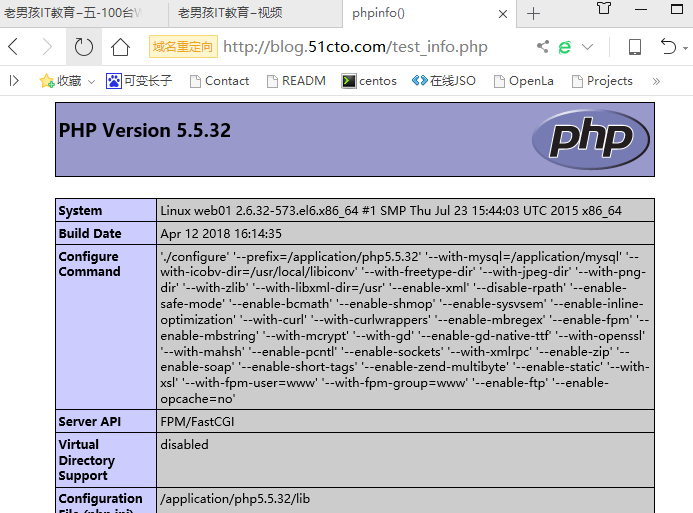
/application/nginx/sbin/nginx -t

/application/nginx/sbin/nginx -s reload

#### 页面访问测试

http://blog.51cto.com/test\_info.php

成功如下图



### 配置PHP连接数据库-进行测试

[root@web01 blog]# cd /application/nginx/html/blog/

[root@web01 blog]# cat test\_mysql.php

<?php

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

if($link\_id){

echo "mysql successful by oldboy !";

}else{

echo mysql\_error();

}

?>

#### 启动nginx

[root@web01 blog]# ../../sbin/nginx -s reload



#### 设置nginx以及php开机自启动

echo "/application/nginx/sbin/nginx" >>/etc/rc.local

echo "/application/php/sbin/php-fpm" >>/etc/rc.local

# wordpress博客搭建实战

[root@web01 tools]# tar xf wordpress-4.9.4-zh\_CN.tar.gz

[root@web01 tools]# cp -a wordpress/\* /application/nginx/html/blog/

[root@web01 tools]# chown -R www.www /application/nginx/html/blog/

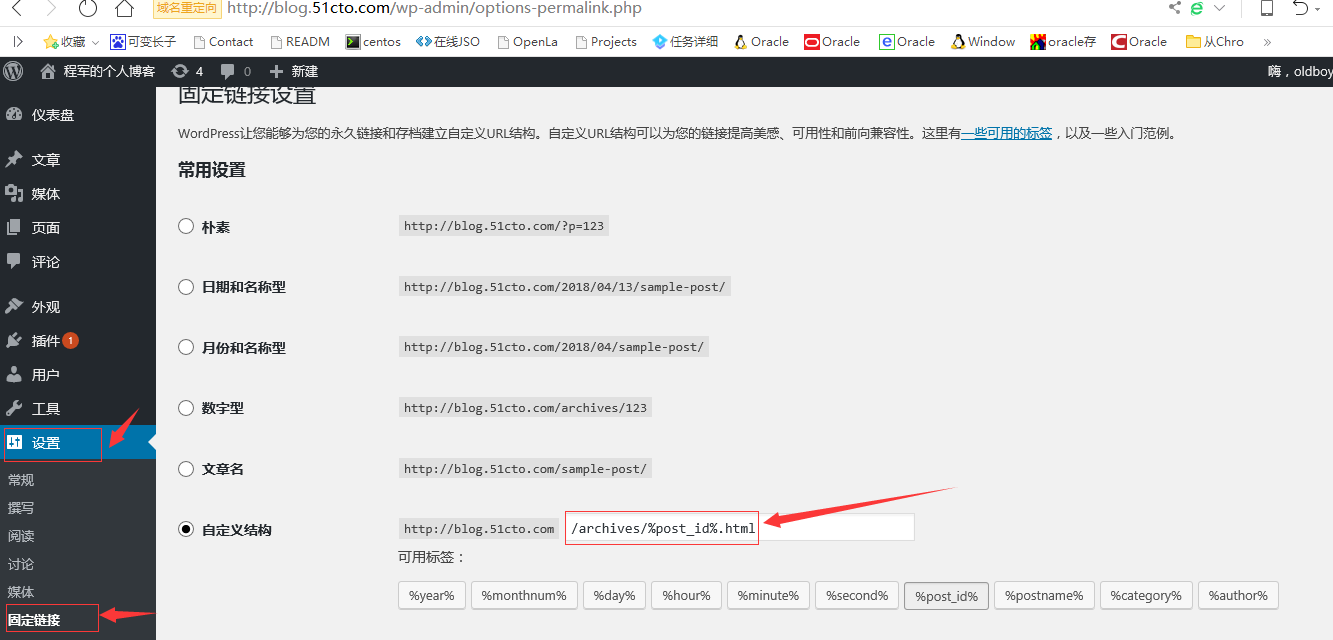
[root@web01 tools]# /application/nginx/sbin/nginx -t

nginx: the configuration file /application/nginx-1.6.3//conf/nginx.conf syntax is ok

nginx: configuration file /application/nginx-1.6.3//conf/nginx.conf test is successful

[root@web01 tools]# /application/nginx/sbin/nginx -s reload

## 实现wordpress博客URL静态化

第一步：首先在后台依次单击设置—-固定链接—–自定义结构，然后输入下面的代码，并保存更改，如下图：  


/archives/%post\_id%.html

### Blog配置文件添加容器：

[root@web01 ~]# cat /application/nginx/conf/extra/blog.conf

server {

listen 80;

server\_name blog.51cto.com;

root html/blog;

index index.php index.html index.htm;

access\_log logs/blog\_access.log;

location /{

if ( -f $request\_filename/index.html){

rewrite (.\*) $1/index.html break;

}

if (-f $request\_filename/index.php){

rewrite (.\*) $1/index.php;

}

if ( !-f $request\_filename){

rewrite (.\*) /index.php;

}

}

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

root html/blog;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

}页面访问

<http://blog.51cto.com/archives/13.html>



## 将web服务器数据库迁移到DB库

1、导出lnmp中Wordpress数据库

[root@web01 tools]# mysqldump -uroot -poldboy123 wordpress -B |gzip>bak.sql.gz

2、将备份库发送到db01

[root@web01 tools]# scp bak.sql.gz root@172.16.1.51:/home/oldboy/tools/

The authenticity of host '172.16.1.51 (172.16.1.51)' can't be established.

RSA key fingerprint is 4b:fb:3b:33:33:a7:31:52:54:dc:d5:54:e9:d4:74:23.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '172.16.1.51' (RSA) to the list of known hosts.

root@172.16.1.51's password:

bak.sql.gz 100% 142KB 142.3KB/s 00:00

3、在db01操作解压gzip备份包

[root@db01 mysql]# cd /home/oldboy/tools/

[root@db01 tools]# gzip -d bak.sql.gz

4、设置db01数据库密码

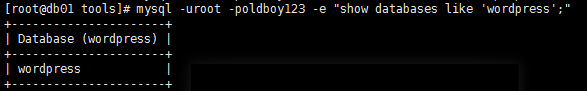
[root@db01 tools]# mysqladmin -u root -p password oldboy123

5、导入数据库

[root@db01 tools]# mysql -uroot -poldboy123 <bak.sql

6、查看表

[root@db01 tools]# mysql -uroot -poldboy123 -e "show databases like 'wordpress';"



[root@db01 tools]# mysql -uroot -poldboy123 -e "use wordpress;show tables;"

+------------------------+

| Tables\_in\_wordpress |

+------------------------+

| old\_commentmeta |

| old\_comments |

| old\_links |

| old\_options |

| old\_postmeta |

| old\_posts |

| old\_term\_relationships |

| old\_term\_taxonomy |

| old\_termmeta |

| old\_terms |

| old\_usermeta |

| old\_users |

+------------------------+

0、在db01上做数据库授权

[root@db01 tools]# mysql -uroot -poldboy123

1、设置访问库及密码

mysql>grant all on wordpress.\* to wordpress@'172.16.1.%' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

2、刷新一下

mysql> flush privileges;

3、查看

mysql> select user,host from mysql.user;

+-----------+------------+

| user | host |

+-----------+------------+

| root | 127.0.0.1 |

| wordpress | 172.16.1.% |

| root | ::1 |

| | db01 |

| root | db01 |

| | localhost |

| root | localhost |

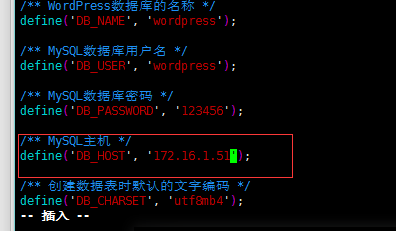
+-----------+------------+

7 rows in set (0.00 sec)

4、在web01上nginx设置db01库的IP地址

[root@web01 blog]#cd /application/nginx/html/blog

[root@web01 blog]# vim wp-config.php



5、进入wordpress库

mysql> use wordpress;

6、查看表项

mysql> show tables;

7、查看表内容 \G为竖立显示

mysql> select \* from old\_posts\G;

## 8、将blog的图片迁移到NFS服务器

原图片目录

[root@web01 uploads]# pwd

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

9、nfs服务器创建账户

[root@nfs01 ~]# useradd -u 501 www

[root@nfs01 ~]# id www

uid=501(www) gid=501(www) 组=501(www)

10、修改nfs配置文件

[root@nfs01 ~]# vim /etc/exports

#share /data/ by cj for zhangkun at 20180329

/data 172.16.1.0/24(rw,sync,all\_squash,anonuid=501,anongid=501)

[root@nfs01 ~]# mkdir /data/nfs-blog

[root@nfs01 ~]# chown -R www.www /data/

11、web01上查看NFS共享文件信息及挂载

[root@web01 uploads]# showmount -e 172.16.1.31

Export list for 172.16.1.31:

/data 172.16.1.0/24

[root@web01 wp-content]# mount -t nfs 172.16.1.31:/data/nfs-blog /application/nginx/html/blog/wp-content/uploads/

[root@web01 wp-content]# df -HP

Filesystem Size Used Avail Use% Mounted on

172.16.1.31:/data/nfs-blog 8.7G 1.7G 6.6G 20% /application/nginx-1.6.3/html/blog/wp-content/uploads

Echo “/bin/mount -t nfs 172.16.1.31:/data/nfs-blog /application/nginx/html/blog/wp-content/uploads/” >>/etc/rc.local

12、移回图片数据

[root@web01 wp-content]# cd uploads/

[root@web01 uploads]# cp -a /tmp/2018/ .

# 带包一键lnmp脚本 赠送项

#一键MYSQL服务器部署

#!/bin/bash

#create mysql db01 scripts

#zuozhe cj

#time 20180413

hostname db01

sed -i 's#backup#db01#' /etc/sysconfig/network

useradd -s /sbin/noligin mysql -M

mkdir –p /application/

mkdir -p /home/oldboy/tools

cd /home/oldboy/tools/

wget https://downloads.mysql.com/archives/get/file/mysql-5.5.49-linux2.6-x86\_64.tar.gz

tar zxf mysql-5.5.49-linux2.6-x86\_64.tar.gz

mv mysql-5.5.49-linux2.6-x86\_64 /application/mysql-5.5.49

cd /application/mysql-5.5.49/

ln -s /application/mysql-5.5.49/ /application/mysql

chown -R mysql.mysql /application/mysql/

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

cd /application/mysql/bin

sed -i 's#/usr/local/#/application/#g' mysqld\_safe

cd ..

sed -i 's#/usr/local/#/application/#g' support-files/mysql.server

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

chmod +x /etc/init.d/mysqld

\cp support-files/my-small.cnf /etc/my.cnf

/application/mysql/bin/mysqld\_safe --user=mysql &

PATH="/application/mysql/bin:$PATH"

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

. /etc/profile

lsof -i :3306

/etc/init.d/mysqld stop

chkconfig mysqld on

#例子-创建wordpress博客数据库##

mysqladmin -u root password 'zchx123'

mysql -uroot -pzchx123

create database wordpress;

#grant all on wordpress.\* to wordpress@'localhost' identified by '123456';

grant all on wordpress.\* to wordpress@'172.16.1.%' identified by '123456';

flush privileges;

#######################################################################################

#一键部署php

#!/bin/bash

#by php server

#zuozhe cj

#time 20180413

##yum安装php依赖库

#更新bash源

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

#更新epel源

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

yum install -y zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel libmcrypt-devel mcrypt mhash

cd /home/oldboy/tools/

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

cd libiconv-1.14

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

make &&make install

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

##上传php-5.5.32.tar.gz包到/home/oldboy/tools/

#####

wget http://mirrors.sohu.com/php/php-5.5.32.tar.gz

tar zxf php-5.5.32.tar.gz

cd php-5.5.32

./configure \

--prefix=/application/php5.5.32 \

--with-mysql=/application/mysql \

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

--with-freetype-dir \

--with-jpeg-dir \

--with-png-dir \

--with-zlib \

--with-libxml-dir=/usr \

--enable-xml \

--disable-rpath \

--enable-safe-mode \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-inline-optimization \

--with-curl \

--with-curlwrappers \

--enable-mbregex \

--enable-fpm \

--enable-mbstring \

--with-mcrypt \

--with-gd \

--enable-gd-native-ttf \

--with-openssl \

--with-mahsh \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-zip \

--enable-soap \

--enable-short-tags \

--enable-zend-multibyte \

--enable-static \

--with-xsl \

--with-fpm-user=nginx \

--with-fpm-group=nginx \

--enable-ftp \

--enable-opcache=no

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

touch ext/phar/phar.phar

make && make install

ln -s /application/php5.5.32/ /application/php

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

cd /application/php/etc/

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

/application/php/sbin/php-fpm

#######################################################################################

#一键Nginx部署

#!/bin/bash

yum install pcre pcre-devel openssl-devel -y

useradd www -s /sbin/nologin -M

mkdir -p /home/data/

mkdir -p /application/

cd /home/data/

wget -q http://nginx.org/download/nginx-1.6.3.tar.gz

tar xf nginx-1.6.3.tar.gz

cd nginx-1.6.3

./configure --user=www --group=www --prefix=/application/nginx1.6.3 --with-http\_stub\_status\_module --with-http\_ssl\_module

make && make install

ln -s /application/nginx-1.6.3/ /application/nginx

/application/nginx/sbin/nginx

mkdir -p /application/nginx/conf/extra

## 一键yum\_lnmp脚本

#!/bin/bash  
#create LNMP scripts  
#zuozhe cj  
#time 20180413  
#######################################################################  
#################一键MYSQL服务器部署#######################################  
###########################################################################  
hostname web01  
/bin/sed -i 's#backup#web01#' /etc/sysconfig/network  
/bin/sed -i 's#IPADDR=10.0.0.100#IPADDR=10.0.0.8#g' /etc/sysconfig/network-scripts/ifcfg-eth0  
/bin/sed -i 's#IPADDR=172.16.1.100#IPADDR=172.16.1.8#g' /etc/sysconfig/network-scripts/ifcfg-eth1  
##########################################################################################  
#配置使用内部YUM仓库172.16.1.61 conf/etc/yum.repos.d/yum.repo####  
#/bin/echo -e "[yum]\n  
#name=Server\n  
#baseurl=http://10.0.0.61\n  
#enable=1\n  
#gpgcheck=0"|egrep -v "^$" >/etc/yum.repos.d/yum.repo  
#/bin/sed -i '18a enabled=0' /etc/yum.repos.d/CentOS-Base.repo  
#/bin/sed -i '26a enabled=0' /etc/yum.repos.d/CentOS-Base.repo  
#/bin/sed -i '34a enabled=0' /etc/yum.repos.d/CentOS-Base.repo  
#yum --enablerepo=yum --disablerepo=base,extras,updates,epel list  
#####安装lnmp必备的基础库及程序  
yum install -y zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libxslt-devel libmcrypt-devel  
yum install -y mcrypt mhash  
yum install gcc nfs-utils rpcbind -y  
yum install pcre pcre-devel openssl-devel -y  
####################################################################  
useradd -s /sbin/noligin mysql -M  
useradd www -s /sbin/nologin -M  
mkdir -p /home/oldboy/tools  
mkdir -p /application/  
mkdir -p /data/nfs-blog  
mkdir -p /server/scripts/  
cd /home/oldboy/tools/  
wget http://10.0.0.61/mysql-5.5.49-linux2.6-x86\_64.tar.gz  
tar zxf mysql-5.5.49-linux2.6-x86\_64.tar.gz  
mv mysql-5.5.49-linux2.6-x86\_64 /application/mysql-5.5.49  
cd /application/mysql-5.5.49/  
ln -s /application/mysql-5.5.49/ /application/mysql  
chown -R mysql.mysql /application/mysql/  
./scripts/mysql\_install\_db --basedir=/application/mysql/ --datadir=/application/mysql/data/ --user=mysql  
cd /application/mysql/bin  
sed -i 's#/usr/local/#/application/#g' mysqld\_safe  
cd ..  
sed -i 's#/usr/local/#/application/#g' support-files/mysql.server  
cp support-files/mysql.server /etc/init.d/mysqld  
chmod +x /etc/init.d/mysqld  
\cp support-files/my-small.cnf /etc/my.cnf  
/application/mysql/bin/mysqld\_safe --user=mysql &  
PATH="/application/mysql/bin:$PATH"  
echo "PATH="/application/mysql/bin:$PATH"" >>/etc/profile  
. /etc/profile  
lsof -i :3306  
/etc/init.d/mysqld stop  
chkconfig mysqld on  
  
#########################################################################  
###########################################################################  
##############一键Nginx部署################################################  
################################################################################  
#\!/bin/bash  
#Web server   
##zuozhe CJ  
#time 20180418  
#yum install pcre pcre-devel openssl-devel -y  
cd /home/oldboy/tools  
/usr/bin/wget http://10.0.0.61/nginx-1.6.3.tar.gz  
tar xf nginx-1.6.3.tar.gz  
cd nginx-1.6.3  
./configure --user=www --group=www --prefix=/application/nginx1.6.3 --with-http\_stub\_status\_module --with-http\_ssl\_module  
make && make install  
ln -s /application/nginx1.6.3/ /application/nginx  
/application/nginx/sbin/nginx  
/bin/mkdir -p /application/nginx/conf/extra  
/bin/mkdir -p /application/nginx/html/{blog,www}  
echo "blog" > /application/nginx/html/blog/index.html  
echo "www" > /application/nginx/html/www/index.html  
cd /application/nginx/conf  
#编写www配置文件  
echo -e "server {\n  
 listen 80;\n  
 server\_name www.51cto.com 51cto.com;\n  
 location / {\n  
 root html/www;\n  
 index index.html index.html;\n  
 rewrite ^/(.\*) http://www.51cto.com/\$1 permanent;\n  
 access\_log logs/www\_access.log main gzip buffer=32k flush=5s;\n  
 }\n  
 location ~.\*\.(php|php5)?\$ {\n  
 root html/www;\n  
 fastcgi\_pass 127.0.0.1:9000;\n  
 fastcgi\_index index.php;\n  
 include fastcgi.conf;\n  
 }\n  
 }"|egrep -v "^$" >extra/www.conf  
#编写默认index配置文件  
echo -e "server {\n  
 listen 80;\n  
 server\_name locahost;\n  
 location / {\n  
 root html/blog;\n  
 index index.php index.htm;\n  
 access\_log logs/index\_access.log;\n   
 }\n  
 }"|egrep -v "^$" >extra/index.conf  
#编写bbs配置文件  
echo -e "server {\n  
 listen 80;\n  
 server\_name bbs.51cto.com;\n  
 location / {\n  
 root html/bbs;\n  
 index index.html index.htm;\n  
 }\n  
 access\_log logs/bbs\_access.log;\n   
 }"|egrep -v "^$" >extra/bbs.conf  
  
#编写blog配置文件带静态化  
echo -e "server {\n  
 listen 80;\n  
 server\_name blog.51cto.com;\n  
 root html/blog;\n  
 index index.php index.html index.htm;\n  
 access\_log logs/blog\_access.log;\n  
location /{\n  
 if ( -f \$request\_filename/index.html){\n  
 rewrite (.\*) \$1/index.html break;\n  
 }\n  
 if (-f \$request\_filename/index.php){\n  
 rewrite (.\*) \$1/index.php;\n  
 }\n  
 if ( \!-f \$request\_filename){\n  
 rewrite (.\*) /index.php;\n  
 }\n  
 }\n  
location ~ .\*\.(php|php5)?\$ {\n  
 root html/blog;\n  
 fastcgi\_pass 127.0.0.1:9000;\n  
 fastcgi\_index index.php;\n  
 include fastcgi.conf;\n  
 }\n  
}"|egrep -v "^$" >extra/blog.conf  
#配置监控状态  
echo -e "server {\n  
 listen 80;\n  
 server\_name status.51cto.com;\n  
 location / {\n  
 stub\_status on;\n  
 access\_log off;\n  
 }\n  
}"|egrep -v "^\$" >extra/status.conf  
#备份配置文件  
cp nginx.conf nginx.conf\_$(date +%F)  
#优化主配置文件  
echo -e "worker\_processes 1;\n  
events {\n  
 worker\_connections 1024;\n  
}\n  
http {\n  
 include mime.types;\n  
 default\_type application/octet-stream;\n  
 sendfile on;\n  
 keepalive\_timeout 65;\n  
 include extra/\*.conf;\n  
 include extra/index.conf;\n  
 include extra/www.conf;\n  
 include extra/blog.conf;\n  
 include extra/bbs.conf;\n  
}"|egrep -v "^$" >nginx.conf  
#nginx日志切割脚本  
echo -e "#\!/bin/bash\n  
DR="/application/nginx/logs/"\n  
IP=172.16.1.41\n  
cd \$DR &&\\\   
mv www\_access.log www\_access\_\$(date +%F -d -1day).log\n  
mv blog\_access.log blog\_access\_\$(date +%F -d -1day).log\n  
mv bbs\_access.log bbs\_access\_\$(date +%F -d -1day).log\n  
mv index\_access.log index\_access\_\$(date +%F -d -1day).log\n  
/application/nginx/sbin/nginx -s reload\n  
##rsync to backup server\n  
rsync -az \$DR --delete rsync\_backup@\$IP::backup/ --password-file=/etc/rsync.passwd\n  
#del date before 7 day ago\n  
find \$DR -type f -name \"\*.log\" -mtime +7 |xargs rm -f"|egrep -v "^$" >/server/scripts/cut\_nginx\_log.sh  
chmod +x /server/scripts/cut\_nginx\_log.sh  
#写入定时任务  
echo "#nginx is log back cut\n00 00 \* \* \* /bin/sh /server/scripts/cut\_nginx\_log.sh >/dev/null 2>&1" >>/var/spool/cron/root  
  
#检查语法与重启  
../sbin/nginx -t  
../sbin/nginx -s reload  
###数据库连接测试文件编写  
echo -e "<?php\n  
 \$link\_id=mysql\_connect('db01','wordpress','123456') or mysql\_error();\n  
 if(\$link\_id){\n  
 echo "mysql successful by oldboy !";\n  
 }else{\n  
 echo mysql\_error();\n  
 }\n  
?>"|egrep -v "^$" >/application/nginx/html/blog/test\_mysql.php  
#######################################################################################  
#例子-创建wordpress博客数据库##可以手动或自动建库，要自动请取消前面注释  
#mysqladmin -u root password 'zchx123'  
#mysql -uroot -pzchx123  
#create database wordpress;  
#grant all on wordpress.\* to wordpress@'localhost' identified by '123456';  
#flush privileges;  
###########################################################################  
#################################################  
#cd /home/oldboy/tools/  
#yum install -y zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libxslt-devel libmcrypt-devel  
#yum install -y mcrypt mhash  
cd /etc/yum.repos.d/  
mkdir tmp -p  
mv CentOS-Base.repo CentOS-Debuginfo.repo CentOS-fasttrack.repo CentOS-Media.repo CentOS-Vault.repo yum.repo tmp/  
wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo  
wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo  
yum install libcurl-devel -y  
rm -rf CentOS-Base.repo epel.repo  
mv tmp/\* .  
rpm -qa zlib-devel libxml2-devel libjpeg-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel libxslt-devel libmcrypt-devel mcrypt mhash  
cd /home/oldboy/tools/  
wget http://10.0.0.61/libiconv-1.14.tar.gz  
tar xf libiconv-1.14.tar.gz  
cd libiconv-1.14  
./configure --prefix=/usr/local/libiconv  
make && make install  
  
##上传php-5.5.32.tar.gz包到/home/oldboy/tools/  
#####  
wget http://10.0.0.61/php-5.5.32.tar.gz  
tar zxf php-5.5.32.tar.gz  
cd php-5.5.32  
ln -s /application/mysql/lib/libmysqlclient.so.18 /usr/lib64/  
touch ext/phar/phar.phar  
make && make install  
#php测试页  
echo "<?php phpinfo(); ?>" >/application/nginx/html/blog/test\_info.php  
ln -s /application/php5.5.32/ /application/php  
cp php.ini-production /application/php/lib/php.ini  
cd /application/php/etc/  
cp php-fpm.conf.default php-fpm.conf  
/application/php/sbin/php-fpm  
  
#######################################################################################################################################  
################wordpress 博客搭建############################################################################################  
#################################################################################################################  
#cd /home/oldboy/tools/  
#tar xf wordpress-4.9.4-zh\_CN.tar.gz  
#cp -a wordpress/\* /application/nginx/html/blog/  
#chown -R www.www /application/nginx/html/blog/  
#/application/nginx/sbin/nginx -t  
#application/nginx/sbin/nginx -s reload  
#########################  
#删除blog-html测试页文件  
\rm -rf/application/nginx/html/blog/index.html  
##nfs挂载####  
/etc/init.d/rpcbind start  
/bin/echo "/etc/init.d/rpcbind start" >> /etc/rc.local  
/bin/mount -t nfs 172.16.1.31:/data/nfs-blog /application/nginx/html/blog/wp-content/uploads/  
/bin/echo "/bin/mount -t nfs 172.16.1.31:/data/nfs-blog /application/nginx/html/blog/wp-content/uploads/" >>/etc/rc.local  
###检查挂载效果  
df -HP  
#########  
#/etc/init.d/network reload  
uname -n