

一、HTTP协议概述

二、HTTP协议版本

三、HTTP方法

GET和POST方法

四、HTTP状态码（面试&排错）

1、成功(2字头)

200 OK

201 Created

202 Accepted

2、重定向(3字头)

301 Moved Permanently

302 Move temporarily

3、请求错误(4字头)

400 Bad Request

401 Unauthorized（路由器）

403 Forbidden

404 Not Found

405 Method Not Allowed

407 Proxy Authentication Required

408 Request Timeout

4、服务器错误(5、6字头)

500 Internal Server Error

502 Bad Gateway

503 Service Unavailable

504 Gateway Timeout

五、HTTP协议请求流程分析

1、当用户输入一个域名地址（URL）

2、首先要进行DNS域名解析：

3、其中会获取到端口号（:80），通过该端口进行通信连接

4、接下来会建立TCP的三次握手：

5、该过程结束过后会实现TCP的四次断开：

6.服务器收到客户端的确认后会立即进入CLOSED关闭状态，结束TCP的连接。

六、HTTP报文格式详解

HTTP 请求发送消息报文格式

Nginx网站服务

LNMP架构及应用部署

1、部署Nginx 软件

1.Nginx脚本：使nginx服务的启动、停止、重载等操作更加方便

2.Nginx安装路径

3.Nginx服务器的主配置文件

4.Nginx配置文件主要分成四部分，每部分包含若干个指令：

5.状态统计模块

6.虚拟主机应用VS虚拟机

7.虚拟主机分类：

基于域名：多个域名解析为一个IP地址，不同域名访问到不同网站内容

基于IP地址：服务器拥有多个IP，不同IP访问不同网站的内容

基于端口：相同IP地址的不同端口访问到不同网站内容

2.安装Mysql数据库

1) 基于源码包安装MySQL

2) 安装后的调整

3) 初始化数据库

4) 设置环境变量

5) 添加系统服务

3、安装PHP解析环境

1) 编译安装php

2) 安装后的调整

3) 为了提高PHP解析效率，建议将相应版本的ZendGuardLander也装上。

4、配置Nginx支持PHP环境

若要让Nginx能够解析PHP网页，有两种方法可以选择；

一、Nginx充当中介

二、通过使用PHP的FPM模块来调用本机的PHP环境

配置Nginx支持PHP解析：

平台测试完成

5、LNMP平台中部署WEB应用

上传bbs代码：

设置权限及模板文件

准备数据库并配置相关授权

为了使访问更方便，对 /usr/local/nginx/conf/nginx.conf做出如下修改

一、HTTP协议概述

- Web服务器所使用的是HTTP协议 -----HTTP服务器
- HTTP 服务器存储因特网中的数据
- 客户端向服务器发送HTTP请求
- 服务器会在HTTP响应中回送所有请求的数据

二、HTTP协议版本

HTTP协议是互联网上应用最广泛的一种网络协议

目的：发布和接收Web服务器上的HTML页面

HTTP协议的版本：

●HTTP 0.9

已过时。只接受GET一种请求方法，没有在通讯中指定版本号，且不支持请求头。

由于该版本不支持POST 方法，所以客户端无法向服务器传递太多信息。

●HTTP 1.0

这是第一个在通讯中指定版本号的HTTP协议版本

至今仍被广泛采用，特别是在代理服务器中

●HTTP 1.1

当前版本

持久连接被默认采用

能很好地配合代理服务器工作

支持以管道方式

同时发送多个请求

以便降低线路负载，提高传输速度

●HTTP 2.0（未实现） 区别于 WEB2.0（互动，已实现）

HTTP 2.0即超文本传输协议2.0，是下一代HTTP协议

https://网址

目的：

- 在开放互联网上增加使用加密技术，
- 以提供强有力的保护去遏制主动攻击

- DANE RFC6698 允许域名管理员不通过第三方CA自行发行证书

三、HTTP方法

1. HTTP支持几种不同的请求命令，这些命令被称为HTTP方法(HTTP method)
2. 每条HTTP请求报文会包含一个方法，告诉服务器要执行什么动作
3. HTTP协议有多种获得Web资源的方法，常用的有两种: GET 和POST

HTTP方法

- **GET** : **不安全** 发送请求报文的时候，是否要进行地址加密；明文密码
- **PUT** : 将资源上传到服务器
- **DELETE** : 删除服务器中的资源
- **POST** : **比较安全** 密文密码
- **HEAD** : 获取响应头部

GET和POST方法

例如访问: `http://www.test.com/a.php?id=123`

GET方法:

请求如果访问正常，我们会从服务器的日志中获取200状态码

POST方法:

我们传递给a.php的id参数依旧是123

但是浏览器的URL将不会显示后面的id=123字样

因此表单类或者有用户名、密码等内容提交时建议使用POST方式

不管使用哪种方式，最终a.php获取到的值是一样的。

GET	POST
从指定的服务器上获取数据	提交数据给指定服务器处理
请求能被缓存	请求不能被缓存
请求会保存在浏览器的浏览记录里	请求不会保存在浏览器的浏览记录里
请求要有长度的限制	请求没有长度的限制
主要用于获取数据，查询的字符串会显示在URL中，不安全	查询的字符串不会显示在URL中，比较安全

四、HTTP状态码（面试&排错）

当使用浏览器访问某一个 URL, 会根据请求URL返回状态码

- 通常整正常的状态码为2xx, 3xx (如200)
- 如果出现异常会返回4xx, 5xx (如404)

1、成功(2字头)

这一类型的状态码，代表请求已成功被服务器接收、理解、并接受。

200 OK

请求已成功

201 Created

请求已经被实现

202 Accepted

服务器已接受请求，但尚未处理

2、重定向(3字头)

301 Moved Permanently

被请求的资源已永久移动到新位置

并且将来任何对此资源的引用，都应该使用本响应返回的若干个URI之一。

```
[root@localhost ~]# curl http://www.360buy.com
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

注意：

对于某些使用HTTP/1.0 协议的浏览器，当它们发送的POST 请求得到了一个301响应的的话，接下来的重定向请求将会变成GET方式

302 Move temporarily

请求的资源临时从不同的URI响应请求。由于这样的重定向是临时的，客户端应当继续向原有地址发送以后的请求。只有在Cache-Control或Expires中进行了指定的情况下，这个响应才是可缓存的。

3、请求错误(4字头)

400 Bad Request

- 1、语义有误，当前请求无法被服务器理解
除非进行修改，否则客户端不应该重复提交这个请求
- 2、请求参数有误

401 Unauthorized (路由器)

- 当前请求需要用户验证
- 验证错误 的问题

403 Forbidden

- 服务器已经理解请求，但是拒绝执行它
- 类似dns,ssh ,tcp wrappers,都有允许及拒绝的策略
- 本机IP在拒绝列表里----地址被拒绝

404 Not Found

- 请求失败，请求所希望得到的资源未被在服务器上发现。
- 路径已切换

405 Method Not Allowed

请求行中指定的请求方法不能被用于请求相应的资源

407 Proxy Authentication Required

- 与401响应类似，客户端必须在代理服务器上身份验证
- 代理服务器必须返回一个Proxy-Authenticate 用以进行身份询问
- 客户端可以返回一个Proxy-Authorization 信息头用以验证

408 Request Timeout

请求超时

4、服务器错误(5、 6字头)

500 Internal Server Error

服务器端的源代码出现错误

502 Bad Gateway

服务器网关或者代理工作的服务器出现错误

503 Service Unavailable

- “双十一” & “春运”
- 由于临时的服务器维护或者过载，服务器当前无法处理请求。
- **注意:503状态码的存在并不意味着服务器在过载的时候必须使用它。**

某些服务器只不过是希望拒绝客户端的连接。

504 Gateway Timeout

服务器网关或者代理服务器尝试执行请求超时

未能及时从上游服务器

(URI 标识出的服务器，例如HTTP、FTP、 LDAP)

或者辅助服务器

(例如DNS)收到响应。

注意:某些代理服务器在DNS查询超时时会返回400或者500错误

错误页面的定义

```
[root@localhost ~]# curl -I http://www.taobao.com
```

```
HTTP/1.1 301 Moved Permanently
```

```
Server: Tengine
```

```
Date: Sun, 29 Mar 2020 07:57:30 GMT
```

```
Content-Type: text/html
```

```
Content-Length: 278
```

```
Connection: keep-alive
```

```
Location: https://www.taobao.com/
```

```
Via: cache11.cn1402[,0]
```

```
Timing-Allow-Origin: *
```

```
EagleId: 65597d1f15854686503608484e
```

淘宝网



抱歉！页面无法访问.....

可能因为：

网址有错误 >请检查地址是否完整或存在多余字符

网址已失效 >可能页面已删除，活动已下线等

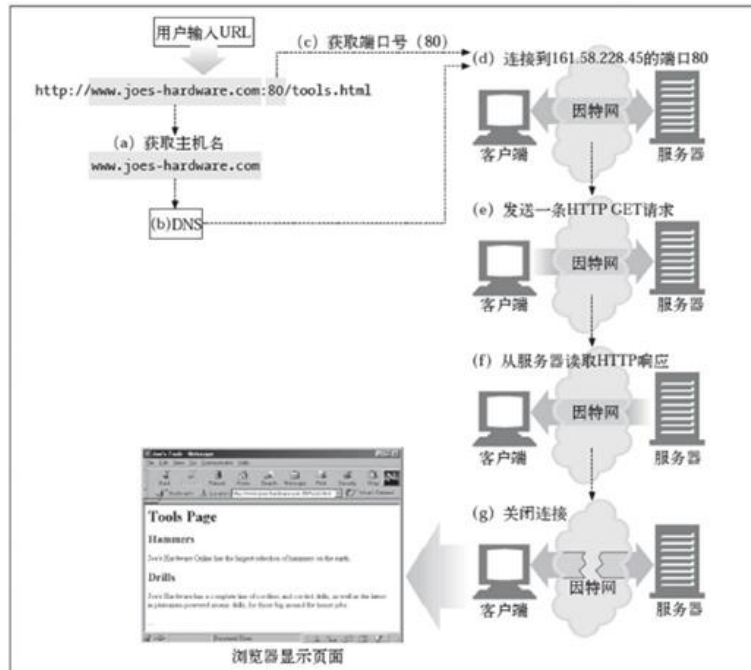
或者逛逛：[淘宝首页](#) | [我的淘宝](#) | [搜宝贝/店铺](#)

系统提示

对不起，你找的文件服务器上找不到！

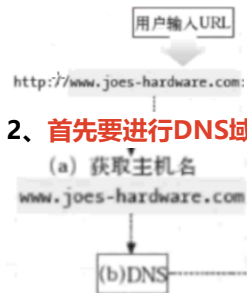
[返回首页](#)

五、HTTP协议请求流程分析



1、当用户输入一个域名地址 (URL)

首先判断它的协议名称，根据该协议名称寻找对应的协议解析方式。



2、首先要进行DNS域名解析:

1. 查找浏览器中的缓存
2. 查找系统中的hosts文件
3. 访问本地的DNS服务器 (LDNS local dns 缓存 202.106.0.20)
4. 通过浏览器访问，LDNS找到根域 (.)
5. 通过已知的根域访问，LDNS找到根域下的顶级域 (.com)
6. 通过该顶级域进行访问，LDNS接下来找到需要的二级域 (.joes-hardware.com)
7. 通过已知的二级域找对应的主机名 (www.joes-hardware.com) 的IP解析记录给LDNS
8. LDNS返回给客户机解析记录及缓存到本地

追踪DNS的解析过程

```
[root@localhost ~]# dig +trace www.baidu.com
; <<<>> DiG 9.9.4-RedHat-9.9.4-61.el7 <<<>> +trace www.baidu.com
;; global options: +cmd
.           511413    IN      NS      d.root-servers.net.
.           511413    IN      NS      f.root-servers.net.
.           511413    IN      NS      m.root-servers.net.
.           511413    IN      NS      c.root-servers.net.
.           511413    IN      NS      k.root-servers.net.
.           511413    IN      NS      b.root-servers.net.
.           511413    IN      NS      j.root-servers.net.
.           511413    IN      NS      a.root-servers.net.
.           511413    IN      NS      i.root-servers.net.
.           511413    IN      NS      l.root-servers.net.
.           511413    IN      NS      h.root-servers.net.
```

```
.          511413  IN   NS   e.root-servers.net.
.          511413  IN   NS   g.root-servers.net.
.          511413  IN   RRSIG  NS 8 0 518400 20200411050000 202003290400
00 33853 . ZrhZdviGE3eP87YHMEYZYO3e6+QhpfqqDH7xqr8B+wnS7cvw8pzmUGgr
M9pr5vS0GLcZ9eS3R+9IP+6H+lbwBU4EcRpe0yuVt2E+FjXXSLf3FMW/
RhCPL0aAv1jLMb/zAkBXjWsqOaxaeNWUEkkdOcdtAUUE03ND7q+PRbSV
PZHr3hvUvtiVSZbIXSV7Hd2n0dXlI734Ru+IAmF0Sf/J7tgZMJzO1gF7
fWgRvYje2BiH5TujRxP3DGb8UjIK/IMUY9xoc8sdiarzXJ/ooqDj7/q2
2dtPmhq2mRSePiVcP028V0IKOeBDRRUjFvbSX85xofPR5jGsDatYmK5B Ws79wg=;; Received 1097 bytes
from 192.168.200.111#53(192.168.200.111) in 3539 ms
# a-m 共13台根域DNS服务器，d.root-servers.net.负责了本次解析
```

```
com.       172800  IN   NS   l.gtld-servers.net.
com.       172800  IN   NS   b.gtld-servers.net.
com.       172800  IN   NS   c.gtld-servers.net.
com.       172800  IN   NS   d.gtld-servers.net.
com.       172800  IN   NS   e.gtld-servers.net.
com.       172800  IN   NS   f.gtld-servers.net.
com.       172800  IN   NS   g.gtld-servers.net.
com.       172800  IN   NS   a.gtld-servers.net.
com.       172800  IN   NS   h.gtld-servers.net.
com.       172800  IN   NS   i.gtld-servers.net.
com.       172800  IN   NS   j.gtld-servers.net.
com.       172800  IN   NS   k.gtld-servers.net.
com.       172800  IN   NS   m.gtld-servers.net.
com.       86400   IN   DS   30909 8 2 E2D3C916F6DEEAC73294E8268FB5885
044A833FC5459588F4A9184CF C41A5766com.      86400   IN   RRSIG  DS 8 1 86400
20200411050000 2020032904000
0 33853 . bOgHpfJv5wYEjWrr9IBIEP1GStStDVq/hFIKhfy52rmhpuA3wGlpeu
ol8CuWJyQ43iB3L+0+cp4gGo3LWJx80+73Eb3FW9ed1PSZa3JVmIBIX9
M5kYd08H7ScGr+ms1fkGhNKKGlyN6/4wvC/m+1BpialrzXOz+9lujepR
DlrAtE2m2jPf99CO7KZFZA0yha0G4gw8/MrSHsRgz3tZS0jVCS8vnb14
R+XP5p4/AjtAjqCL+E/xvk7Prx+CjiDWMCMWjcm/AtZSwZRC3ltYkiRH
WnkOuMU9Alc4inMGGtzWkgrZqPN1Hckp4ZJpDA13bqxzRJ3mbsUGpmXB NHpyWg=;; Received 1173
bytes from 192.203.230.10#53(e.root-servers.net) in 7137 ms
# 解析com 顶级域的共有13台DNS服务器，l.gtld-servers.net.负责了本次解析
```

```
baidu.com. 172800  IN   NS   ns2.baidu.com.
baidu.com. 172800  IN   NS   ns3.baidu.com.
baidu.com. 172800  IN   NS   ns4.baidu.com.
baidu.com. 172800  IN   NS   ns1.baidu.com.
baidu.com. 172800  IN   NS   ns7.baidu.com.
CK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN NSEC3 1 1 0 -
CK0Q1GIN43N1ARRC9OSM6QPQR81H
5M9A NS SOA RRSIG DNSKEY NSEC3PARAMCK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN RRSIG
NSEC3 8 2 86400 20200403045024 20200
327034024 56311 com. c6uPvcETse+c/D/ZkGPqZKIjwJ4z+XjCD7UjBR12FuskCK2Uo1jBx7F5
YIAcE+Peh5eQkX5qrTEkK/X5x2PnBxfGpW8txZ+J76nwreB96VZg2fld
tGndAnEda8/ohuJHmnsPDzdY6D8uVZC9FbTvSCKYVLMouZWPuPICqDeR
BlkpklucF5/eEtXkMaF258HlnTGrUwrwB2wwwzqLklqTFUg==HPVUNU64MJQUM37BM3VJ6O2UBJCHOS00.com.
86400 IN NSEC3 1 1 0 - HPVVN3Q5E5GOQP2QFE2LEM4SVB9C
0SJ6 NS DS RRSIGHPVUNU64MJQUM37BM3VJ6O2UBJCHOS00.com. 86400 IN RRSIG NSEC3 8 2 86400
20200402042254 20200
326031254 56311 com. BMGkv/EhNgYC2gENaLUjIFQ6Ft31V/4RImxUOhv8XSTHV5F3z5XRUUDb
sDmtYAAxDxGXjrLCMk3f6Z8XM25TSdqUswqTjDUjUdviGgcAxO6n8f8yH
R9ph7DeCjj8ScGKC6C5zj628BwMaFml28vB9o8pgusb0grxkFQlv7fFx
5irWwcpWYDhxxzJohUNvywdhOoG0DHqNgkpGON2MZ3YI+g=;; Received 761 bytes from
192.52.178.30#53(k.gtld-servers.net) in 379 ms
# 解析baidu.com.的有5台服务器，ns2.baidu.com.负责了本次解析
```

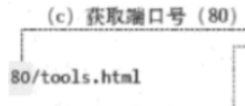


```

www.baidu.com.      1200      IN      CNAME  www.a.shifen.com.
a.shifen.com.       1200      IN      NS     ns4.a.shifen.com.
a.shifen.com.       1200      IN      NS     ns3.a.shifen.com.
a.shifen.com.       1200      IN      NS     ns2.a.shifen.com.
a.shifen.com.       1200      IN      NS     ns1.a.shifen.com.
a.shifen.com.       1200      IN      NS     ns5.a.shifen.com.
;; Received 239 bytes from 202.108.22.220#53(ns1.baidu.com) in 31 ms

```

3、其中会获取到端口号 (:80) , 通过该端口进行通信连接



4、接下来会建立TCP的三次握手:

1. 客户端的一方会发送一个SYN (seq=x) 的请求报文, 此时SYN的值为1, 并进入SYN_SENT状态, 等待服务器确认。
2. 服务器接收到该请求报文后, 如果同意进行连接的话会返回一个值为1的ACK (seq=x+1) 的确认报文; 同时也会发送一个值为1的SYN (seq=y) 请求报文, 因为数据的建立是双向的, 此时服务器进入SYN_RECV状态。
3. 客户端接收到后会返回一个ACK (ack=y+1) 确认报文, 返回值也为1。此时客户端和服务器双方进入ESTABLISHED (TCP连接成功) 状态, 完成三次握手, 通过该端口建立了通信连接。

(d) 连接到161.58.228.45的端口80



1. 当客户端发送一条HTTP GET 请求后

(e) 发送一条HTTP GET请求



2. 服务器会读取该请求并进行相应的HTTP响应

(f) 从服务器读取HTTP响应

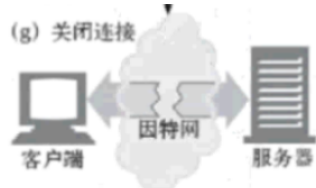


5、该过程结束过后会实现TCP的四次断开:

1. 客户端发送一个FIN=1 (seq=u) 的释放连接报文信息, 此时, 客户端进入FIN-WAIT-1 (终止等待1) 状态
2. 服务器接收到后会返回一个值为1的ACK (ack=u+1) 确认报文, 此时, 服务端就进入了CLOSE-WAIT (关闭等待) 状态。
3. 客户端收到服务器的确认请求后, 此时, 客户端就进入FIN-WAIT-2 (终止等待2) 状态, 等待服务器发送连接释放报文 (在这之前还需要接受服务器发送的最后的的数据)。
4. 服务器将最后的数据发送完毕后, 就向客户端发送连接释放报文FIN=1 (ack=u+1), 并附带一个自己的序列号 (seq=w), 此时, 服务器就进入了LAST-ACK (最后确认) 状态, 等待客户端的确认。
5. 客户端收到连接释放报文后, 发出确认报文ACK=1 (ack=w+1), 而自己的序列号是seq=u+1, 此时, 客户端就进入了TIME-WAIT (时间等待) 状态。

但是TCP连接还没有完全释放，客户端必须经过2*MSL（最长报文段寿命）的等待时间后，才能进入CLOSED状态。

6.服务器收到客户端的确认后会立即进入CLOSED关闭状态，结束TCP的连接。



此时客户端已经获得了该IP地址的内容，如果想要进行更多内容的获取，则会再次进行三次握手和四次断开的过程，从而获得相应的内容。



六、HTTP报文格式详解

HTTP 请求发送消息报文格式

一次完整的http 请求消息包括：

- 一个请求行
- 若干消息头以及实体内容
- 消息头和实体内容可以没有，消息头和实体内容之间有一个空行

```
[root@localhost ~]# curl -I http://www.taobao.com
HTTP/1.1 301 Moved Permanently
Server: Tengine
Date: Sun, 29 Mar 2020 07:57:30 GMT
Content-Type: text/html
Content-Length: 278
Connection: keep-alive
Location: https://www.taobao.com/
Via: cache11.cn1402[,0]
Timing-Allow-Origin: *
EagleId: 65597d1f15854686503608484e
您在 /var/spool/mail/root 中有新邮件
[root@localhost ~]# curl -I http://www.taobao.com/jbluigyoglgk1564893135
```

Request Headers		[Raw]	[Header Definitions]
GET /jinkou.htm HTTP/1.1			
Client			
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9			
Accept-Encoding: gzip, deflate			
Accept-Language: zh-CN,zh;q=0.9			
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36			
Cookies			
Cookie			
Miscellaneous			
Referer: http://www.lingshi.com/			
Security			
Upgrade-Insecure-Requests: 1			
Transport			
Connection: keep-alive			
Host: www.lingshi.com			
Response Headers		[Raw]	[Header Definitions]
HTTP/1.1 200 OK			
Cache			
Date: Sun, 29 Mar 2020 08:56:45 GMT			
Vary: Accept-Encoding,User-Agent			
Entity			
Content-Encoding: gzip			
Content-Length: 4227			
Content-Type: text/css			
Last-Modified: Mon, 07 Jan 2019 04:08:52 GMT			
Miscellaneous			
Accept-Ranges: bytes			
Server: Apache/2.2.17 (Unix) DAV/2 PHP/5.3.5			
Transport			
Connection: Keep-Alive			
Keep-Alive: timeout=5, max=100			

```
[root@localhost ~]# curl -I http://www.baidu.com/
HTTP/1.1 200 OK
Accept-Ranges: bytes
```

Cache-Control: private, no-cache, no-store, proxy-revalidate, no-transform
Connection: keep-alive
Content-Length: 277
Content-Type: text/html
Date: Sun, 29 Mar 2020 09:44:46 GMT
Etag: "575e1f60-115"
Last-Modified: Mon, 13 Jun 2016 02:50:08 GMT
Pragma: no-cache
Server: bfe/1.0.8.18 # 隐藏服务器名, 做到保护的作用

Nginx网站服务

[web服务器--nginx简介](#)

nginx 介绍

Nginx (engine x) 是一个高性能的HTTP和反向代理web服务器，同时也提供了IMAP/POP3/SMTP服务。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，在BSD-like 协议下发行。

特点

- 占有内存少
- 并发能力强

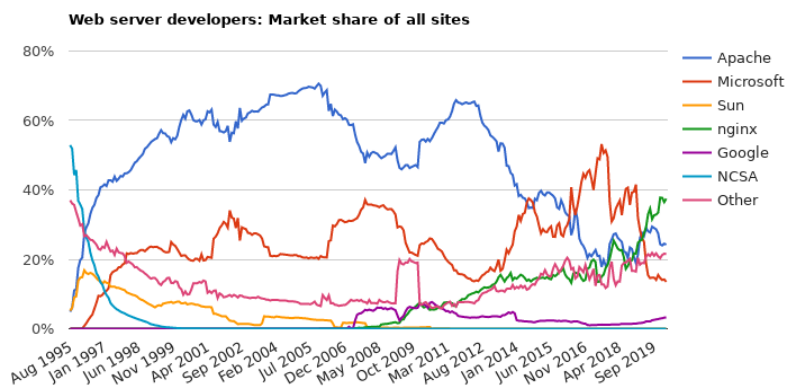
事实上nginx的并发能力确实在同类型的网页服务器中表现较好

中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等

Nginx代码完全用C语言从头写成，已经移植到许多体系结构和[操作系统](#)，包括：[Linux](#)、[FreeBSD](#)、[Solaris](#)、[Mac OS X](#)、[AIX](#)以及[Microsoft Windows](#)。

Nginx有自己的函数库，并且除了[zlib](#)、[PCRE](#)和[OpenSSL](#)之外，标准模块只使用系统C库函数。

而且，如果不需要或者考虑到潜在的授权冲突，可以不使用这些第三方库。



ps:

参考现在服务器相关数据

<http://news.netcraft.com/>

当然[google趋势](#)也是一个可选得备案。

有一个数据统计更全面的:

<http://w3techs.com/>

统一编程语言使用率的网站:

<https://www.tiobe.com/tiobe-index/>

<http://redmonk.com/>

发展历程

Nginx是由伊戈尔·赛索耶夫为俄罗斯访问量第二的Rambler.ru站点（俄文：Рамблер）开发的，第一个公开版本0.1.0发布于2004年10月4日。

其将源代码以类BSD许可证的形式发布，

因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。

2011年6月1日，nginx 1.0.4发布。

Nginx 可以在大多数 UnixLinux OS 上编译运行，并有 Windows 移植版。

Nginx 的1.4.0稳定版已经于2013年4月24日发布，

一般情况下，对于新建站点，建议使用最新稳定版作为生产版本，已有站点的升级紧迫性不高。

在连接高并发的情况下，Nginx是Apache服务不错的替代品：

Nginx在美国是做虚拟主机生意的老板们经常选择的软件平台之一。

能够支持高达 50,000 个并发连接数的响应，感谢Nginx为我们选择了 epoll and kqueue作为开发模型。

社区版

Many websites are under additional load due to COVID-19. You can help, and we want to help you.
[Get access to free resources at nginx.com](#)

nginx: download

Mainline version → **测试版**

[CHANGES](#) [nginx-1.17.9](#) [pgp](#) [nginx/Windows-1.17.9](#) [pgp](#)

Stable version → **稳定版**

[CHANGES-1.16](#) [nginx-1.16.1](#) [pgp](#) [nginx/Windows-1.16.1](#) [pgp](#)

Legacy versions → **旧版本**

CHANGES-1.14	nginx-1.14.2 pgp	nginx/Windows-1.14.2 pgp
CHANGES-1.12	nginx-1.12.2 pgp	nginx/Windows-1.12.2 pgp
CHANGES-1.10	nginx-1.10.3 pgp	nginx/Windows-1.10.3 pgp
CHANGES-1.8	nginx-1.8.1 pgp	nginx/Windows-1.8.1 pgp
CHANGES-1.6	nginx-1.6.3 pgp	nginx/Windows-1.6.3 pgp
CHANGES-1.4	nginx-1.4.7 pgp	nginx/Windows-1.4.7 pgp
CHANGES-1.2	nginx-1.2.9 pgp	nginx/Windows-1.2.9 pgp
CHANGES-1.0	nginx-1.0.15 pgp	nginx/Windows-1.0.15 pgp
CHANGES-0.8	nginx-0.8.55 pgp	nginx/Windows-0.8.55 pgp
CHANGES-0.7	nginx-0.7.69 pgp	nginx/Windows-0.7.69 pgp
CHANGES-0.6	nginx-0.6.39 pgp	
CHANGES-0.5	nginx-0.5.38 pgp	

Source Code

NGINX
english
[русский](#)

[news](#)
[about](#)
[download](#)
[security](#)
[documentation](#)
[faq](#)
[books](#)
[support](#)

[trac](#)
[twitter](#)
[blog](#)

[unit](#)
[njs](#)

公司版

March 2020 Web Server Status: X nginx news X NGINX | High Performance L... X

nginx.com

产品 论坛 DNS权威转移服务 付费版 社区 资源展示 帮助 语言 服务团队列表

NGINX PRODUCTS SOLUTIONS RESOURCES SUPPORT PRICING BLOG [FREE TRIAL](#) [CONTACT US](#)

You want to do good. NGINX is here to help.

Access Free Resources for
Websites Impacted by COVID-19

[LEARN MORE](#)

What brought you to nginx.com today?

www.shang.png 全屏显示

淘宝Tengine : <http://tengine.taobao.org/>

简介

Tengine是由淘宝网发起的Web服务器项目。它在Nginx的基础上，针对大访问量网站的需求，添加了很多高级功能和特性。Tengine的性能和稳定性已经在大型的网站如**淘宝网**，**天猫商城**等得到了很好的检验。它的最终目标是打造一个高效、稳定、安全、易用的Web平台。

从2011年12月开始，Tengine成为一个开源项目，Tengine团队在积极地开发和维护着它。Tengine团队的核心成员来自于**淘宝**、**搜狗**等互联网企业。Tengine是社区合作的成果，我们欢迎大家**参与其中**，贡献自己的力量。

[下载](#)

分析网站后台软件产品

<https://netcraft.app/browser/?r=toolbar>

最新的稳定版: [nginx-1.16.1](#) [pgp](#)

最新的开发版: [nginx-1.17.9](#) [pgp](#)



LNMP架构及应用部署.docx
639.46KB

LNMP架构及应用部署

LNMP也称为LEMP，E来自于Nginx的发音[engine x]

LNMP就像构建LAMP平台一样，构建LNMP平台也需要用到Linux服务器，MySQL数据库，PHP解释环境等应用。

P（PHP、Perl、Python）

wamp是什么？xampp

1、部署Nginx 软件

1) 安装支持软件:

Nginx的配置及运行需要pcre、zlib等软件包的支持，因此应预先安装这些软件的开发包（devel），以便提供相应的库和头文件，确保Nginx的安装顺利完成。

```
[root@localhost ~]# systemctl stop firewalld
```

```
[root@localhost ~]# iptables -F
```

```
[root@localhost ~]# setenforce 0
```

```
[root@localhost ~]# yum -y install pcre-devel zlib-devel openssl-devel
```

2) 创建运行用户、组:

Nginx服务程序默认以nobody身份运行，建议为其创建专门的用户账号，以便更准确地控制其访问权限，增加灵活性、降低安全风险。如：创建一个名为nginx的用户，不建立宿主目录，也禁止登录到shell环境。

```
[root@localhost ~]# useradd -M -s /sbin/nologin nginx
```

3) 编译安装nginx:

释放nginx源码包

```
[root@localhost ~]# tar xf nginx-1.16.0.tar.gz -C /usr/src/
```

4) 配置编译:

```
[root@localhost ~]# cd /usr/src/nginx-1.16.0/
[root@localhost nginx-1.16.0]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --
with-http_stub_status_module --with-http_ssl_module --with-http_flv_module --with-
http_gzip_static_module && make && make install
```

注：配置前可以参考：./configure --help给出说明

- --prefix 设定Nginx的安装目录
- --user和--group 指定Nginx运行用户和组
- --with-http_stub_status_module 启用http_stub_status_module模块以支持状态统计
- --with-http_ssl_module 启用SSL模块
- --with-http_flv_module 启用FLV模块，提供寻求内存使用基于时间的偏移量文件

为了使Nginx服务器的运行更加方便，可以为主程序nginx创建链接文件，以便管理员直接执行nginx命令就可以调用Nginx的主程序。

```
[root@localhost nginx-1.16.0]# ln -s /usr/local/nginx/sbin/nginx /usr/local/bin/
[root@localhost nginx-1.16.0]# ll /usr/local/bin/nginx
lrwxrwxrwx 1 root root 27 3月 29 19:45 /usr/local/bin/nginx -> /usr/local/nginx/sbin/nginx
```

5) Nginx的运行控制:

与Apache的主程序httpd类似，Nginx的主程序也提供了“-t”选项用来对配置文件进行检查，以便找出不当或错误的配置。配置文件nginx.conf默认位于安装目录/usr/local/nginx/conf/目录中。若要检查位于其他位置的配置文件，可使用“-c”选项来指定路径。

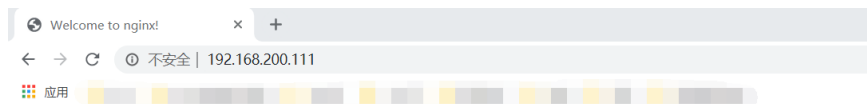
```
[root@nginx conf]# nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

6) 启动、停止Nginx:

直接运行nginx即可启动Nginx服务器，这种方式将使用默认的配置文件，若要改用其他配置文件，需添加“-c 配置文件路径”选项来指定路径。需要注意的是，若服务器中已安装有httpd（apache）等其他WEB服务软件，应采取措施（修改端口，停用或卸载其他软件）避免部突。

```
[root@nginx conf]# netstat -anpt |grep :80
[root@nginx conf]# nginx
[root@nginx conf]# netstat -anpt |grep :80
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN      6810/nginx:
master
```

通过检查 Nginx程序的监听状态，或者在浏览器中访问此WEB服务（默认页面将显示“Welcome to nginx!”），可以确认Nginx服务是否正常运行。



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

主程序Nginx支持标准的进程信号，通过kill或者killall 命令传送 0

- HUP 重载配置等同于 -1
- QUIT 退出进程等同于 -3
- KILL 杀死进程等同于 -9

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

```
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     11013/nginx: master
```

```
[root@localhost ~]# killall -QUIT nginx
```

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

```
[root@localhost ~]# nginx
```

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

```
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     11478/nginx: master
```

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

```
access.log  error.log  nginx.pid
```

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

```
11478
```

```
[root@localhost ~]# kill -QUIT $(cat /usr/local/nginx/logs/nginx.pid)
```

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

```
[root@localhost ~]# nginx
```

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

```
tcp        0      0 0.0.0.0:80          0.0.0.0:*          LISTEN     11563/nginx: master
```

1.Nginx脚本：使nginx服务的启动、停止、重载等操作更加方便

```
[root@localhost ~]# vim /etc/init.d/nginx
```

```
#!/bin/bash
```

```
#chkconfig:2345 99 20
```

```
#服务在2345级别启动，第99个启动，第20个关闭
```

```
#description:nginx server control script
```

```
#$0是脚本名
```

```
PROG="/usr/local/nginx/sbin/nginx"
```

```
#nginx的绝对路径
```

```
PIDF="/usr/local/nginx/logs/nginx.pid"
```

```
case "$1" in
```

```
start)
```

```
    $PROG
```

```
;;
```

```

stop)
    kill -s QUIT $(cat $PIDF)
;;
restart)
    $0 stop
    $0 start
;;
reload)
    kill -s HUP $(cat $PIDF)
;;
*)
    echo " Usage:$0{start|stop|restart|reload}"
    exit 1
esac
exit 0

```

```
#####
```

```

[root@localhost ~]# vim /etc/init.d/nginx
[root@localhost ~]# chmod +x /etc/init.d/nginx
[root@localhost ~]# chkconfig --add nginx
# 添加到系统服务当中
[root@localhost ~]# systemctl start nginx
[root@localhost ~]# systemctl restart nginx
[root@localhost ~]# systemctl reload nginx
[root@localhost ~]# chkconfig --list nginx
nginx          0:关    1:关    2:开    3:开    4:开    5:开    6:关

```

```

[root@localhost ~]# rpm -qa | grep bind
bind-libs-lite-9.9.4-61.el7.x86_64
bind-9.9.4-61.el7.x86_64
bind-libs-9.9.4-61.el7.x86_64
rpcbind-0.2.0-44.el7.x86_64
keybinder3-0.3.0-1.el7.x86_64
bind-utils-9.9.4-61.el7.x86_64
bind-license-9.9.4-61.el7.noarch

```

```

[root@localhost ~]# vim /etc/named.conf

```

```

options {
    directory "/var/named";

```



```
};  
zone "liruifang.com" IN {  
    type master;  
    file "liruifang.com.zheng";  
};
```

```
[root@localhost ~]# vim /var/named/liruifang.com.zheng
```

```
$TTL 86400
```

```
@      SOA      liruifang.com.      admin.liruifang.com.      (  
        20200329  
        3H  
        15M  
        1W  
        1D  
)
```

```
      IN      NS      ns.liruifang.com.  
ns     IN      A       192.168.200.111  
www    IN      A       192.168.200.111
```

```
[root@localhost ~]# chgrp named /var/named/liruifang.com.zheng
```

```
[root@localhost ~]# vim /etc/resolv.conf
```

```
nameserver 192.168.200.111
```

```
[root@localhost ~]# systemctl start named
```

```
#这里有可能需要用restart
```

```
[root@localhost ~]# nslookup www.liruifang.com
```

```
Server:      192.168.200.111
```

```
Address:     192.168.200.111#53
```

```
Name:  www.liruifang.com
```

```
Address: 192.168.200.111
```

elinks是linux上的浏览器

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

```
[root@localhost ~]# elinks --dump http://www.liruifang.com
```

```
Welcome to nginx!
```

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [1]nginx.org.

Commercial support is available at [2]nginx.com.

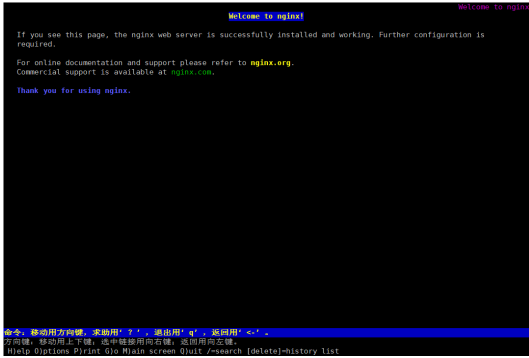
Thank you for using nginx.

References

Visible links

1. <http://nginx.org/>
2. <http://nginx.com/>

或者: [root@localhost ~]# lynx http://www.liruifang.com



又或者: [root@localhost ~]# curl http://www.liruifang.com

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

2.Nginx安装路径

```
[root@localhost ~]# ls /usr/local
bin  etc  games  include  lib  lib64  libexec  nginx  sbin  share  src
[root@localhost ~]# ls /usr/local/nginx
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  uwsgi_temp
[root@localhost ~]# ll /opt/soft/nginx    #有公司的安装路径不同也是正常的
```

3.Nginx服务器的主配置文件

```
[root@localhost ~]# ll /opt/soft/nginx
[root@localhost ~]# ls /usr/local/nginx/conf/nginx.conf
/usr/local/nginx/conf/nginx.conf
```

4.Nginx配置文件主要分成四部分，每部分包含若干个指令：

- main（全局设置）
- server（主机设置）
- upstream（上游服务器设置，主要为反向代理、负载均衡相关配置）
- location（URL匹配特定位置后的设置）

main部分设置的指令将影响其它所有部分的设置；

server部分的指令主要用于指定虚拟主机域名、IP和端口；

upstream的指令用于设置一系列的后端服务器，设置反向代理及后端服务器的负载均衡；location部分用于匹配网页位置（比如，根目录“/”，“/images”，等等）。

他们之间的关系式：

server继承main，

location继承server；

upstream既不会继承指令也不会被继承。它有自己的特殊指令，不需要在其他地方的应用。

全局配置：

user nginx [nginx]; //运行用户，默认为nobody.

Worker_processes 2; //指定nginx启动的工作进程数量，建议按照cpu数目来指定，和CPU核心数相等。

worker_cpu_affinity 0000000001;0000000010 //为每个进程分配cpu核心程分配到两个cpu,当然可以写多个，或者将一个进程分配到多个cpu

worker_rlimit_nofile 102400; //这个指令是指当一个nginx进程打开的最多文件数目，于

论值应该是最多打开文件数(ulimit -n) 与nginx进程数相除，但是ngeinx分配请求并不是那么均匀，所以最好与ulimit -n的值保持一致。（通过“limit -n数值”可以修改的最多文件数目）

Ps:调系统的文件数目

```
[root@localhost ~]# ulimit -n
```

1024

```
[root@localhost ~]# ulimit -n 102400
```

```
[root@localhost ~]# ulimit -n
```

102400

error logog/erlog; //全局错误日志文件的位置。

pid logs/nginx.pid; //PID文件的位置e

I/O事件配置:

使用“events { }”界定标记, 用来指定Nginx进程的I/O响应模型, 每个进程的连接数等设置

```
events {  
    use epoll;    //使用epoll模型, 对于2.6以上的内核, 建议使用epoll 模型以提高性能  
    worker_connections 4096;    //每个进程允许的最多连接数(默认为1024), 每个进程的连接数应根据实际需要来定,  
    一般在10000以下, 理论上每台nginx服务器的最大连接数为worker_processes*worker_connections, 具体还要看服  
    务器的硬件、带宽等。  
}
```

HTTP配置

使用“http{ }”界定标记, 包括访问日志、HTTP端口、网页目录、默认字符集、连接保持、及虚拟主机、PHP解析等一系列设置。其中大部分配置语句包含在子界定标记“server { }”内

```
http {  
    Include      mime.types;  
    default_type  application/octet-stream;  
    log_format main 'Sremote. addr . Sremote user [$time local]"Srequest" '  
    '$status $body_ bytes. sent"Shttp_ referer" '  
    "$http user agent" "$http. x forwarded for";  
    access_log logs/access.log main;    //访问日志位  
    Sendfile      on;    //支持文件发送(下载)  
    keepalive_timeout 65;    //连接保持超时。
```

#一个server 就是一个虚拟主机

```
server {    //web服务的监听配置e  
    Listen 80;    //监听地址及端口(IP: PORT)  
    server_name www.liruifang.com;    //网站名称(FQDN)  
    Charset utf-8;    //网页的默认字符集  
  
    location/{    //跟目录配置  
        root    html;    //网站根目录的位置安装位置的html中  
        index index.html index.htm;    //默认首页 (索引页)。  
    }
```

```
error_page 500 502503 504 /50x.html;    //内部错误的反 馈页面  
location= /50x.html {    //错误页面配置  
    root    html;  
}  
  
}  
  
}
```

排错经过:

1. 查看IP是否正确
2. 重启服务systemctl restart nginx

3. 查看网络接口状态 Netstat -lnpt
4. Systemctl stop nginx & systemctl start nginx
5. 查看进程状态 Ps aux | grep nginx
6. 更换浏览器测试（浏览器缓存数据）
7. 注意更改数据的位置

5.状态统计模块

```

xxx xx;
xxx xx;
xxx xx;
xxx xx;

events {
    xxx xx;
    xxx xx;
}
HTTP {
    server {
        location {

        }
    }
    location {

    }
}

server {
    location {

    }
    location {

    }
}

```

Nginx内置了HTTP_STUB_ STATUS状态统计模块，用来反馈当前的WEB访问情况。

配置编译参数时可添加—with-http stub stastus module来启用此模块。

要使用Nginx的状态统计功能，除了启用内建模块以外, 还需要修改nginx.conf文件, 指定访问位置并打开stub status配置。

在http配置的servern子配置内添加如下配置项：

重新加载服务：

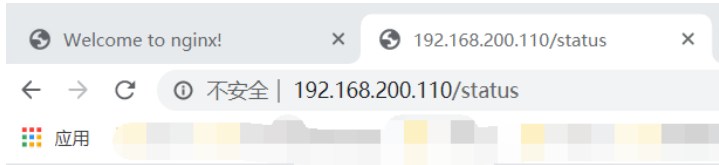
```

[root@localhost ~]# killall -9 nginx
[root@localhost ~]# nginx

```

```
location /status {
    stub_status on;
    access_log off;
}
```

浏览器访问 <http://192.168.200.110/status>



```
Active connections: 2
server accepts handled requests
3 3 13
Reading: 0 Writing: 1 Waiting: 1
```

Active connections表示当前活跃的连接数

第三行的三个数字表示Nginx当前总共处理了3个连接，成功创建3次握手，总共处理了13个请求

Reading表示Nginx读取到客户端Header信息数

Writing表示Nginx返回给客户端的Header信息数

Waiting表示Nginx已经处理完，正在等候下一次请求指令时的驻留连接数

6.虚拟主机应用VS虚拟机

虚拟机：在一台机器上体验多台机器

虚拟主机：

比如对外提供网站，叫这个网站叫站点，买一台服务器且搭建好nginx服务，对外提供业务，一个站点无法将整个服务器的资源承载，无法完全利用CPU、内存，

无法用完全部的资源，在一台机器上装一台nginx软件，且提供多个站点，叫虚拟主机。

7.虚拟主机分类：

基于域名：多个域名解析为一个IP地址，不同域名访问到不同网站内容

<http://www.a.com> <http://www.b.com>

基于IP地址：服务器拥有多个IP，不同IP访问不同网站的内容

<http://192.168.1.1> <http://192.168.1.2>

基于端口：相同IP地址的不同端口访问到不同网站内容

<http://192.168.1.1:80> <http://192.168.1.2:81>

使用Nginx搭建虚拟主机服务器时，每个虚拟WEB站点拥有独立的“server 0”配置段，各自监听的IP地址、端口号可以单独指定，当然网站名称也是不同的。

案例：

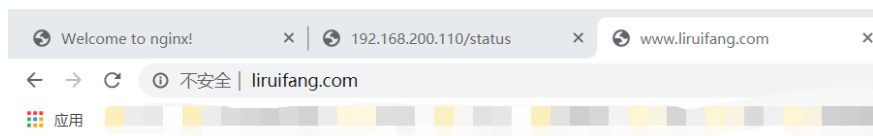
要创建两个站点www.liruifang.com和www.doud.com

为两个虚拟WEB主机分别建立根目录，并准备测试首页。

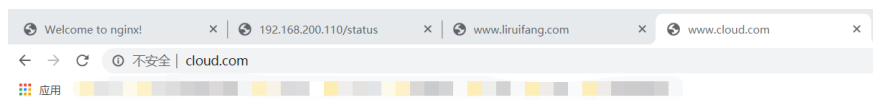
当网页测试不成功时，

杀服务: killall -9 nginx

重启服务: nginx



liruifang test



cloud test

2.安装Mysql数据库

LNMP也称为LEMP, E来自于Nginx的发音[engine x]

LNMP: Linux Nginx Mysql Php

LAMP :Linux Apache MySQL PHP

LNMP就像构建LAMP平台一样, 构建LNMP平台也需要用到Linux服务器,

MySQL 数据库, PHP解释环境等应用。P (PHP、Perl、Python)

wamp是什么? windows apache MySQL PHP

网络营销专业会下载软件: xampp 软件开发专用软件

1) 基于源码包安装MySQL

检查是否有rpm格式的安装包

```
[root@nginx ~]# rpm -e mysql mysql-server mariadb mariadb-server --nodeps
```

#安装的时候不用敲--nodeps

ncurses-devel是cmake的依赖包

```
[root@nginx ~]# yum -y install ncurses-devel
```

```
[root@nginx ~]# rpm -q ncurses-devel
```

ncurses-devel-5.9-14.20130511.el7_4.x86_64

建议采用yum安装方式

```
[root@nginx ~]# yum -y install cmake
```

```
[root@nginx ~]# rpm -q cmake
```

cmake-2.8.12.2-2.el7.x86_64

创建运行用户

```
[root@nginx ~]# useradd -M -s /sbin/nologin mysql
```

解包, 配置, 编译, 安装

```
[root@nginx ~]# tar xf mysql-5.7.24.tar.gz -C /usr/src/
```

```
[root@nginx ~]# cd /usr/src/mysql-5.7.24/
```

```
[root@nginx mysql-5.7.24]# cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci -DWITH_EXTRA_CHARSETS=all -DSYSCONFDIR=/etc && make && make
```

install

- -DCMAKE_INSTALL_PREFIX=/usr/local/mysql //数据库程序安装目录
- -DDEFAULT_CHARSET=utf8 //指定字符集编码
- -DDEFAULT_COLLATION=utf8_general_ci //默认的字符集校对规则, utf8_general_ci适用于utf-8字符集的通用规则
- -DWITH_EXTRA_CHARSETS=all //指定额外支持的字符集编码
- -DSYSCONFDIR=/etc //指定配置文件存放目录

报错处理:

```
-----CMake Error at
cmake/boost.cmake:81 (MESSAGE):
You can download it with -DDOWNLOAD_BOOST=1 -DWITH_BOOST=<directory>

This CMake script will look for boost in <directory>. If it is not there,
it will download and unpack it (in that directory) for you.

If you are inside a firewall, you may need to use an http proxy:

export http_proxy=http://example.com:80

Call Stack (most recent call first):
cmake/boost.cmake:238 (COULD_NOT_FIND_BOOST)
CMakeLists.txt:507 (INCLUDE)

-- Configuring incomplete, errors occurred!
See also "/usr/src/mysql-5.7.24/CMakeFiles/CMakeOutput.log".
See also "/usr/src/mysql-5.7.24/CMakeFiles/CMakeError.log".
-----
```

解决办法:

a. 在/usr/local下创建一个名为boost的文件夹

```
[root@nginx ~]# mkdir /usr/local/boost
```

b. 进入目录并下载boost

```
[root@nginx ~]# cd /usr/local/boost
```

```
[root@nginx boost]# wget
```

```
https://sourceforge.net/projects/boost/files/boost/1.59.0/boost_1_59_0.tar.gz
```

c. 解压boost

```
[root@nginx boost]# tar xf boost_1_59_0.tar.gz
```

d. 继续cmake, 添加上红色部分

```
[root@nginx ~]# cd /usr/src/mysql-5.7.24/
```



```
[root@nginx mysql-5.7.24]# cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DDEFAULT_CHARSET=utf8 -
DDEFAULT_COLLATION=utf8_general_ci -DWITH_EXTRA_CHARSETS=all -DSYSCONFDIR=/etc -
DWITH_BOOST=/usr/local/boost && make && make install
```

2) 安装后的调整

对数据库目录进行权限设置

```
[root@nginx ~]# cd /usr/local/mysql/
[root@nginx mysql]# chown -R mysql:mysql ./
```

建立配置文件 (CentOS7系统默认支持MariaDB数据库，系统默认的/etc/my.cnf配置文件是MariaDB的配置文件)

```
[root@nginx mysql]# vim /etc/my.cnf
```

```
[mysqld]
```

```
datadir=/usr/local/mysql/data
```

```
socket=/tmp/mysql.sock
```

```
[mysqld_safe]
```

```
log-error=/usr/local/mysql/data/mysql.log
```

```
pid-file=/usr/local/mysql/data/mysql.pid
```

3) 初始化数据库

```
[root@nginx mysql]# ./bin/mysqld --user=mysql --basedir=/usr/local/mysql --
datadir=/usr/local/mysql/data --initialize
```

2018-12-08T01:51:39.798903Z 1 [Note] A temporary password is generated for root@nginx: **TvC:Rm1Z1xtG**

- --basedir=/usr/local/mysql/ //指定安装目录 (产品目录)
- --datadir=/usr/local/mysql/data //指定数据目录
- --user=mysql //指定用户身份

4) 设置环境变量

```
[root@nginx mysql-5.7.24]# echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile
```

或者: [root@localhost mysql]# vim /etc/profile

不能保证重定向内容一定能正确输入时，最好用vim编辑，不要直接用echo

```
[root@nginx mysql-5.7.24]# . /etc/profile = source /etc/profile
```

5) 添加系统服务

```
[root@localhost mysql-5.7.24]# cp support-files/mysql.server /etc/init.d/mysqld
```

```
[root@localhost mysql-5.7.24]# chmod +x /etc/init.d/mysqld
```

```
[root@localhost mysql-5.7.24]# /etc/init.d/mysqld start
```

Starting MySQL.Logging to '/usr/local/mysql/data/mysql.log'.

SUCCESS!

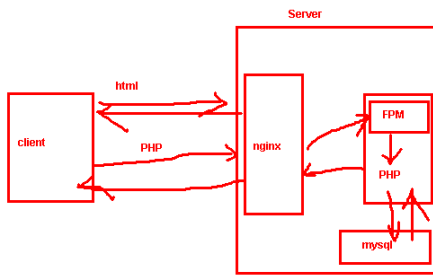
后期修改数据库用户的密码:

```
[root@nginx ~]# mysqladmin -u root -p'TvC:Rm1Z1xtG' password '123456'
```

```
[root@localhost ~]# mysql -uroot -p123456      #登录
```

3、安装PHP解析环境

较新版本（如5.6）的PHP已经自带FPM(fastCGI process manager, FastCGI进程管理器)模块，用来对PHP解析实例进行管理，优化解析效率，因此在配置PHP编译选项时应添加“--enable-fpm”以启用此模块。



1) 编译安装php

```
[root@nginx ~]# yum -y install gd libxml2-devel libjpeg-devel libpng-devel
[root@nginx ~]# tar xf php-5.6.39.tar.gz -C /usr/src/
[root@nginx ~]# cd /usr/src/php-5.6.39/
[root@nginx php-5.6.39]# ./configure --prefix=/usr/local/php5 --with-gd --with-zlib --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysq_config --with-config-file-path=/usr/local/php5 --enable-mbstring --enable-fpm --with-jpeg-dir=/usr/lib && make && make install
```

2) 安装后的调整

```
[root@nginx php-5.6.39]# cp php.ini-production /usr/local/php5/php.ini
[root@nginx php-5.6.39]# ln -s /usr/local/php5/bin/* /usr/local/bin/
[root@nginx php-5.6.39]# ln -s /usr/local/php5/sbin/* /usr/local/sbin/
```

3) 为了提高PHP解析效率，建议将相应版本的ZendGuardLander也装上。

安装ZendGuardLander

(注意：若是64位系统，该软件得到其官网下载64位的相应软件包，若用32位的就会报错。下载地址：

<http://www.zend.com/en/products/guard/downloads>)

```
[root@nginx ~]# tar xf zend-loader-php5.6-linux-x86_64_update1.tar.gz -C /usr/src/
[root@nginx ~]# cd /usr/src/zend-loader-php5.6-linux-x86_64/
[root@nginx zend-loader-php5.6-linux-x86_64]# cp ZendGuardLoader.so /usr/local/php5/lib/php/
```

```
[root@nginx zend-loader-php5.6-linux-x86_64]# vim /usr/local/php5/php.ini
zend_extension=/usr/local/php5/lib/php/ZendGuardLoader.so
zend_loader.enable=1
```

4、配置Nginx支持PHP环境

若要让Nginx能够解析PHP网页，有两种方法可以选择；

一、Nginx充当中介

将访问PHP页面的WEB请求转交给其他服务器（LAMP）去处理；

二、通过使用PHP的FPM模块来调用本机的PHP环境

如果选用FPM方式，则需要先启动php-fpm进程，以便监听PHP解析请求。

参考范例

建立php-fpm.conf配置文件，

并修改其中的PID文件、运行用户、服务数（进程数量）等相关设置，

然后启动php-fpm程序即可（默认监听本机的9000端口）

```
[root@nginx ~]# cd /usr/local/php5/etc/
[root@nginx etc]# cp php-fpm.conf.default php-fpm.conf
[root@nginx etc]# useradd -M -s /sbin/nologin php
[root@nginx etc]# vim php-fpm.conf
25 pid = run/php-fpm.pid           //确认pid文件位置
149 user = php                     //运行用户
150 group = php                    //运行组
246 pm.start_servers = 20          //启动时开启的进程数
251 pm.min_spare_servers = 5      //最少空闲进程数
256 pm.max_spare_servers = 35     //最大空闲进程数
241 pm.max_children = 50          //最多空闲进程数

[root@nginx etc]# /usr/local/sbin/php-fpm    #启动服务
[root@nginx etc]# netstat -anpt |grep php-fpm
tcp        0      0 127.0.0.1:9000      0.0.0.0:*           LISTEN      23027/php-
fpm.conf)
```

在 php-fpm.conf 文件中，pid 配置行指出了PID信息的存放位置，对应的实际路径为/usr/local/php5/var/run/php-fpm.pid，根据上述信息，可以修改Nginx服务脚本，以便在启动/停止Nginx服务器时将php-fpm进程也自动启动/停止。

```
[root@nginx etc]# vim /etc/init.d/nginx
#!/bin/bash
# chkconfig: 2345 99 20
# description: Nginx Server Control Script
PROG="/usr/local/nginx/sbin/nginx"
PIDF="/usr/local/nginx/logs/nginx.pid"
PROG_FPM="/usr/local/sbin/php-fpm"
PIDF_FPM="/usr/local/php5/var/run/php-fpm.pid"
case "$1" in
start)
    $PROG
    $PROG_FPM
;;
stop)
    kill -s QUIT $(cat $PIDF)
    kill -s QUIT $(cat $PIDF_FPM)
;;
restart)
    $0 stop
    $0 start
;;
reload)
    kill -s HUP $(cat $PIDF)
;;
```

```

*)
    echo "Usage: $0 (start|stop|restart|reload)"
    exit 1
esac
exit 0

```

这样，一旦启动或关闭nginx服务，php-fpm程序也会随之启动或关闭，不需要额外再启动或关闭php-fpm。

配置Nginx支持PHP解析：

无论是将PHP页面交给LAMP服务器去解析，
还是调用本机的php-fpm进程进行解析，
都需要在“server{}”配置段中添加location设置，
以便指定当访问.php页面时采取何种操作。

第一种方法：使用nginx的反向代理功能（转交给其他WEB服务器处理）

例如：交给IP地址为192.168.200.112的LAMP服务器处理，
从而实现nginx负责静态，LAMP负责动态页面的分离效果

第二种方法（调用本机的php-fpm进程），使用的配置语句如下所示。

在conf/目录下的fastcgi.conf文件中已经包含必需的宏设置，

可通过include语句添加进来。

```

[root@nginx etc]# vim /usr/local/nginx/conf/nginx.conf
server {
..... //省略部分信息
    location / {
        root    html;
        index  index.php index.html index.htm;
    }
    location ~ \.php$ {                                //访问php页面的配置段
        root html;                                       //PHP网页文档根目录
        fastcgi_pass 127.0.0.1:9000; //php-fpm的监听地址
        fastcgi_index index.php;                       //PHP首页文件
        include fastcgi.conf;                          //包括fastcgi.conf样本配置
    }
}

```

本文档中我选择的是第二种方法，完成修改后重新加载nginx服务即可生效。

```

[root@nginx etc]# killall -HUP nginx

```

在PHP文档根目录下创建一个测试网页，以便测试PHP语言能否正常解析，以及能否访问MySQL数据库。测试页内容如下：

```

[root@localhost ~]# vim /usr/local/nginx/html/php.php
[root@nginx ~]# cat /usr/local/nginx/html/php.php
<?php
phpinfo();

```

?>

PHP Version 5.6.39	
System	Linux localhost 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64
Build Date	Apr 3 2020 14:29:00
Configure Command	'./configure' '--prefix=/usr/local/php5' '--with-gd' '--with-zlib' '--with-mysql=/usr/local/mysql' '--with-mysqli=/usr/local/mysql/bin/mysqli_config' '--with-config-file-path=/usr/local/php5' '--enable-mbstring' '--enable-fpm' '--with-jpeg-dir=/usr/lib'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/php5
Loaded Configuration File	/usr/local/php5/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

平台测试完成

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

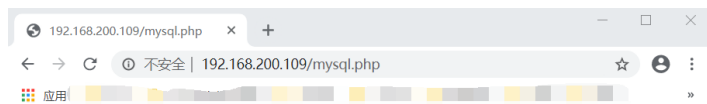
```
<?php
```

```
$link=mysqli_connect('localhost','root','123456'); //连接mysql数据库
```

```
if($link) echo "<h1>恭喜你，大功告成！！</h1>"; //连接成功则返回信息
```

```
mysqli_close(); //关闭数据库连接
```

```
?>
```



恭喜你，大功告成！！

5、LNMP平台中部署WEB应用

LNMP平台与LAMP平台是非常相似的，区别主要在于所用WEB服务软件的不同，而这与使用PHP开发的WEB应用程序并无太大关系，因此PHP应用的部署方法也是类似的。

部署ComsenzDiscuz BBS论坛系统

Discuz!社区论坛是一个采用PHP与MySQL等多种数据库构建的性能优异，功能全面的且安全稳定的社区论坛（BBS）软件，其官方网站为<http://www.discuz.net>

```
[root@nginx ~]# unzip ComsenzDiscuz-DiscuzX-master.zip
```

```
[root@nginx ~]# cd DiscuzX/
```

```
[root@nginx DiscuzX]# ls upload/
```

```
admin.php api.php config crossdomain.xml favicon.ico group.php index.php m misc.php portal.php
```

```
search.php static uc_client
```

api archiver connect.php data forum.php home.php install member.php plugin.php robots.txt source
template uc_server

上传bbs代码:

```
[root@nginx DiscuzX]# mv upload/ /usr/local/nginx/html/bbs
```

浏览器访问 <http://192.168.200.109/bbs/install/index.php>



设置权限及模板文件

```
[root@nginx ~]# cd /usr/local/nginx/html/bbs/config/  
[root@nginx config]# cp config_global_default.php config_global.php  
[root@nginx config]# cp config_ucenter_default.php config_ucenter.php
```

```
[root@nginx ~]# cd /usr/local/nginx/html/bbs  
[root@nginx bbs]# chmod -R 777 config/ data/ uc_client/ uc_server/  
#工作中哪个缺权限给哪个，不要全部给权限，要不太不安全了
```



准备数据库并配置相关授权

```
[root@nginx ~]# mysql -uroot -p123456
```

```
mysql> create database bbs;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> grant all on bbs.* to 'bbs'@'localhost' identified by '123456';
```

```
Query OK, 0 rows affected, 1 warning (0.07 sec)
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.00 sec)
```

Discuz! 安装向导

192.168.200.109/bbs/install/index.php?step=3&install_...

应用

Discuz! 安装向导

Discuz!X3.4 简体中文 UTF8 版 20180101

3. 安装数据库

正在执行数据库安装

检查安装环境

设置运行环境

创建数据库

安装

填写数据库信息

数据库服务器:

localhost

数据库服务器地址, 一般为 localhost

数据库名:

bbs

数据库用户名:

bbs

数据库密码:

123456

数据表前缀:

pre_

同一数据库运行多个论坛时, 请修改前缀

系统信箱 Email:

admin@admin.com

用于发送程序错误报告

填写管理员信息

管理员账号:

admin

管理员密码:

管理员密码不能为空

重复密码:

管理员 Email:

admin@admin.com

下一步

Discuz! 安装向导

Discuz!X3.4 简体中文 UTF8 版 20180101

Discuz! 应用中心

应用中心特意为您准备了一批优秀应用, 插件、模板应有尽有, 无限制扩充站点功能, 建站必备。
快来应用中心装个应用吧!

12.12

狂欢继续8折优惠

DZ论坛小程序仅售 2240

Hybird模板仅售 1584

360 专注微信应用

拼团 砍价 交友

助力 抽奖 投票

维清双12感恩回馈

全场应用 68 折

文章

自动采集发布

西瓜手机应用

50多款 任你挑

SEO百度熊掌号推送

安装: 228

★★★★★

顶象无感验证码

安装: 5,867

★★★★★

手机注册V手机登录

安装: 3,617

★★★★★

Discuz! 短信通

安装: 2.3万

★★★★★

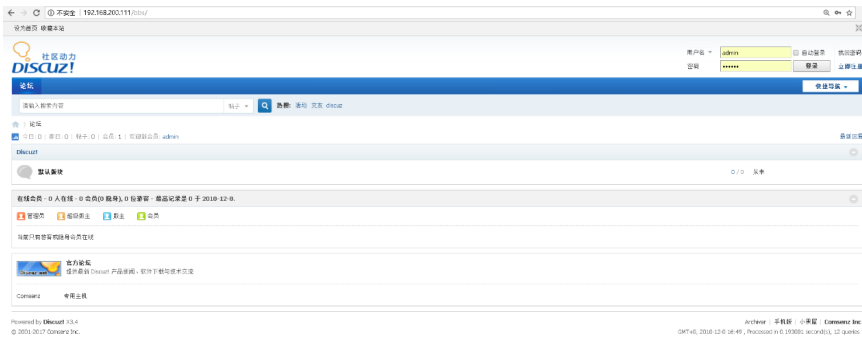
【同盾】论坛防灌水

安装: 1.7万

★★★★★

您的论坛已完成安装, 点此访问

©2001 - 2017 Comsenz Inc.



为了使访问更方便，对 `/usr/local/nginx/conf/nginx.conf` 做出如下修改

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

user  nginx nginx;
worker_processes  2;
error_log  logs/error.log;
pid        logs/nginx.pid;
worker_rlimit_nofile 102400;

events {
    use epoll;
    worker_connections  4096;
}

http {
    include        mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  logs/access.log  main;
    sendfile    on;
    keepalive_timeout  65;

    server {
        listen      80;
        server_name  localhost;
        charset utf-8;

        location / {
            root    html/bbs;
            index  index.php index.html index.htm;
        }
    }
}
```

```
location ~ /\.php$ {
    root html/bbs;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    include fastcgi.conf;
}

location /status {
    stub_status on;
    access_log off;
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}
}

[root@nginx etc]# killall -HUP nginx
```

浏览器访问

<http://192.168.200.109/forum.php?mod=viewthread&tid=1&extra=>

