目录：

[1、网络基础 2](#_Toc418499242)

2、nginx服务19

3、MySQL数据库44

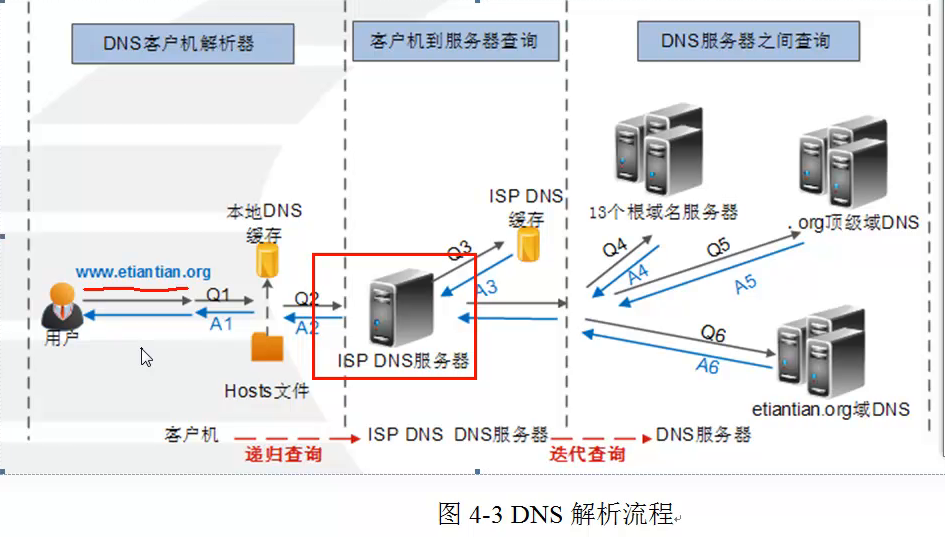
4、FastCGI介绍53

5、安装PHP 54

6、安装wordpress博客61

# 

# 1、网络基础



**HTTP协议简介：**

HTTP协议，全称hypertext transfer protocol，中文名称超文本传输协议，是互联网中最常用的一种网络协议，HTTP的重要应用就是WWW服务。设计HTTP协议的最初目的就是提供一种发布和接收HTML（一种页面标记语言）页面的方法。

HTTP协议是互联网上的通信 协议方案之一，它有很多的应用，但最流行的就是用于web浏览器和web服务器之间的通信应用，即WWW应用或web应用。

WWW，全称world wide web，简称web，中文译为“万维网”。它是目前互联网上最受用户欢迎的信息服务形式。HTTP协议的WWW服务应用的默认服务端口为80，另外一个加密的WWW服务应用（HTTPS）的默认服务端口为443，用于网银，支付等和钱相关的业务。当今，http服务、www服务、web服务，三者的概念已经混淆了。

HTTP的版本从诞生之初到现在经历了几个主要版本：http/0.9，http/1.0，http/1.1，其中，http/1.0是第一个得到广泛使用的版本，而http/1.1为当前使用的主流版本。

**在Linux系统中，使用curl –I 加IP命令可以查询HTTP版本号。**

[root@nfs ~]# curl -I www.baidu.com

HTTP/1.1 200 OK

Date: Mon, 16 Nov 2015 01:21:03 GMT

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

Connection: Keep-Alive

Vary: Accept-Encoding

Set-Cookie: BAIDUID=A7381ED75F0EBAEA7B7F4E47C65922A2:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: BIDUPSID=A7381ED75F0EBAEA7B7F4E47C65922A2; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: PSTM=1447636863; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: BDSVRTM=0; path=/

Set-Cookie: BD\_HOME=0; path=/

Set-Cookie: H\_PS\_PSSID=11194\_1451\_17636\_17620\_17901\_12825\_17782\_17927\_17970\_17833\_17001\_17073\_15595\_11644\_18017; path=/; domain=.baidu.com

P3P: CP=" OTI DSP COR IVA OUR IND COM "

Cache-Control: private

Cxy\_all: baidu+8f5bdaf454fd6a7b4b4eff0ed0121352

Expires: Mon, 16 Nov 2015 01:20:11 GMT

X-Powered-By: HPHP

Server: BWS/1.1

X-UA-Compatible: IE=Edge,chrome=1

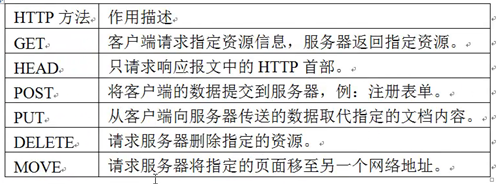
BDPAGETYPE: 1

BDQID: 0xf92dcf1900035887

BDUSERID: 0

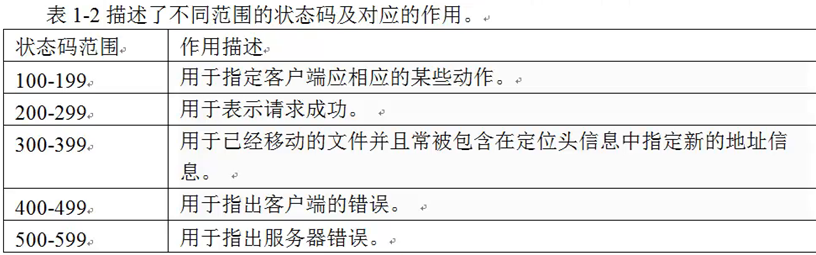
**HTTP请求方法**

在http通信中，每个http请求报文都包含一个方法。用以告诉web服务器端需要执行什么具体的动作，这些动作包括：获取指定web页面、提交内容到服务器、删除服务器上资源文件等，这些http请求报文中包含的方法被称为http方法。其中，常用的http方法见下表：



**HTTP状态码**

http状态码（http status code）是用来表示web服务器相应http请求状态的数字代码，每当web客户端向web服务器发送一个http请求时，web服务器都会返回一个状态响应代码。这个状态码是一个三位数字代码，作用是告知web客户端此次的请求是否成功，或者是采取其他动作方式。



|  |  |
| --- | --- |
| **状态代码** | **详细描述说明** |
| 200 OK | 服务器成功返回页面，这是成功的http请求。返回标准状态码。 |
| moved permanently | 永久跳转到被设定的新位置。 |
| 403 forbidden | 禁止访问。这个请求是合法的，但是服务器端因为匹配了预先设置的规则而拒绝相应客户端的请求。 此问题，一般为服务器权限配置不当导致。 |
| 404 not found | 服务器找不到客户端请求的指定页面。可能是客户端请求了服务器不存在的资源导致。 |
| 500 internal server error | 内部服务器错误。服务器遇到了意料不到的情况，不能完成客户请求。这是比较笼统的报错。一般为服务器设置或者内部程勋问题导致。例如。Selinux开启，而又没有为http设置规则许可，客户访问就是500. |
| 502 bad gateway | 坏的网关，一般是代理服务器请求是，后端服务不可用，或者没有完成相应网关服务器，一般为代理.（代理向下找服务器，服务器不给回应。导致代理回复用户502错误。） |
| 503 service unavailable | 服务器当前不可用，可能因为服务器超载或者停机维护导致，或者代理服务器后面没有可供服务的节点。 |
| 504 gateway timeout | 网关超时，一般网关代理服务器请求后端服务器是，后端服务器没有在特定时间内完成处理请求，一般是服务器过载导致没有在指定时间内放回数据代理服务器。 |

**HTTP报文**

http报文中有很多行内容，不同行的字段都是一些ASCII码串，各个字段的长度也是不同的。http报文有两种，一种是从web客户端发往web服务器的http报文，称为请求报文（request message）。另外一种是从web服务器发往web客户端的报文，称为响应报文（response message），http请求和响应报文的格式类似。

http请求报文（request message）介绍

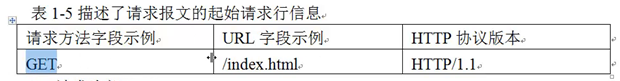
http请求报文由请求行、请求头部（header）、空行和请求报文主体几个部分组成，下表给出了请求报文的一般格式。



下面对http请求报文的每个部分逐一阐述：

1、请求行

请求行是请求报文的第一行，用来说明客户端想要做什么，内容由请求方法字段、URL字段和http协议版本字段共三个字段组成，他们之间用空格分隔。例如，GET/index.html HTTP/1.1



2、请求头部

请求头部由关键字/值对组成，每行一对，关键字和值用英文冒号“：”分隔，请求头部的作业是客户端把请求的相关信息高速给服务器：



与请求报文相关的最常用的请求头是content-TYPE和content-length。

3、空行

最后一个请求头部信息之后是一个空行，发送回车符合换行符，通知web服务器空行以下不会有请求头部的信息了。

4、请求报文主体

请求主体中包括了要发送给web服务器的数据信息；请求报文主体不会应用于http的GET命令方法，而是应用于post方法。post方法适用于需要客户填写表单的场合。

HTTP响应报文（response message）介绍

http响应报文由起始行、响应头部（header）、空行和响应报文主体几个部分组成，和http请求报文格式类似。



下面对响应报文的每个部分逐一阐述：

1、起始行

响应报文的起始行，也叫状态行，用来说明服务器响应客户端请求的状况。一般为协议及版本号，数字状态码、状态情况。例：http/1.1 200 OK

2、响应头部

和请求报文类似，接着起始行的后面一般有若干个头部字段，每个头部字段都包含一个名字和一个值，两者之间用冒号分隔。头部结尾也是以一个空行结束。常见的头部信息有：

content-TYPE；text/html；charset=utf-8

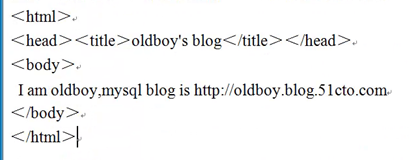
content-length：78

3、空行

最后一个响应头部信息之后是一个空行，发送回车符和换行符，通知客户端空行下文无头部信息了。

4、响应报文主体

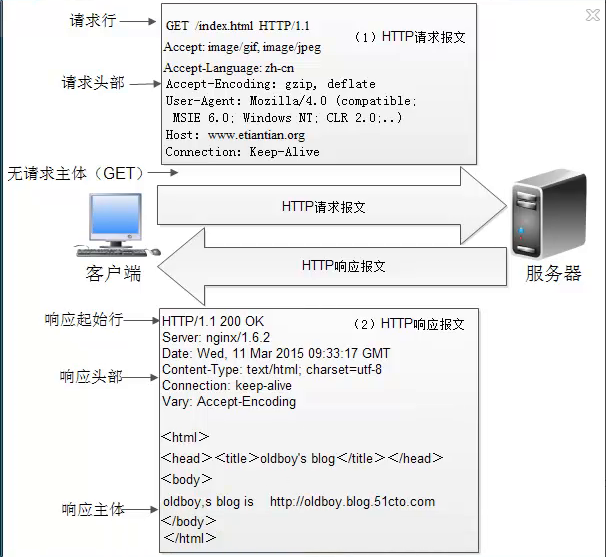
响应报文主体中装在了要返回给客户端的数据。这些数据可以是文本及二进制的（如图片、视频），下面是响应的HTML格式文本数据实例。



一个简单的请求报文和应答报文示例：

web客户端发送一条http请求报文，请求资源http：//www.etiantian.org/index.html，请求报文的起始行中有一个GET命令，资源名称为/index.html。使用的是HTTP/1.1协议，由于请求的方法为GET，即请求报文不需要有主体，因为从服务器是上GET一个简单的页面不需要再请求报文的主体中发送请求数据。

web服务器接受到请求后，返回一条HTTP响应报文。这条响应报文中包含了HTTP的版本号（HTTP/1.1）、成功的状态码（200）、状态描述（OK），以及若干响应头部字段，在所有这些内容之后跟着包含了所请求资源的响应主体。content-length首部说明了响应主体的长度，content-TYPE首部说明了文档的mime类型。



**HTTP协议原理**

HTTP协议属于 OSI模型中的第七层应用层协议, HTTP协议的重要应用就是WWW 服务应用, 下面就以WWW服务应用为例介绍 HTTP协议的通信原理, HTTP协议进行通信时,需要有客户端即终端用户,和服务端即 Web服务器,当 Web客户端向 Web服务器发送请求报文之前,先要通过 TCP/IP协议在 web客户端和服务器之间建立一个 TCP/IP连接。整个http协议请求的工作流程原理步骤如下:

(1) 终端客户在 Web浏览器地址本兰输入访同地址 http://www.etiantian.org/index.htm1;

(2) Web浏览器请求 DNS服务器把域名www.etiantian.org转換成 Web服务器的 IP地址, 此处的解析过程就是DNS解析的原理流程,上文己经讲过了,此处不在累述。

(3) Web浏览器将端口号(默认是80)从访同地址(uRL)中解析出来;

(4) Web浏览器通过解析后的 IP地址及端口号与 Web服务器之间建立一条 TCP连接;

(5) 建立 TCP连接后, Web浏览器向 Web服务器发送一条 HTTP请求报文,请求报文内容格式及信息细节,上文HTTP报文一节己经讲过了,此处不在累述。

(6) Web服务器响应并读取浏览器的请求信息,并返叫一条HTTP响应报文,响应报文内容格式及信息细节,前文也己经讲过了,此处不在累述。

（7）web服务器关闭http连接，关闭TCP连接，web浏览器显示访问的网站内容到屏幕。

上述就是http协议通信原理过程，整个通信原理的重要知识点有：

1、用户访问网站的流程。

2、NDS解析流程细节。

3、建立TCP连接后发送HTTP报文的过程。

4、HTTP请求报文的细节。

5、HTTP相应报文的细节。

6、web服务器请求后端集群存储的细节。

**HTTP协议重点知识小结**

结论: DNS解析原理, http协议原理是高新面试的重点,高级运维必会知识。

1、http协议位于OSI模型中第7层应用层。

2、http协议的重要应用是www服务.

3、用户上网流程, DNS解析原理流程。

4、DNS解析获取到 IP后,建立 TCP連接, 然后发送 http请求细节和服务器响应组节。

5、HTTP请求报文与 HTTP响应报文知识.

6、到送 HTTP服务后情求后端集群节点的流程,nginx---fastcgi—PHP（数据库，存储等）。

7、老男孩推荐书籍, a、 hltp权威指南(前4章) 、b、TCP/IP协议卷一。

HTTP资源

媒体（MIME）类型

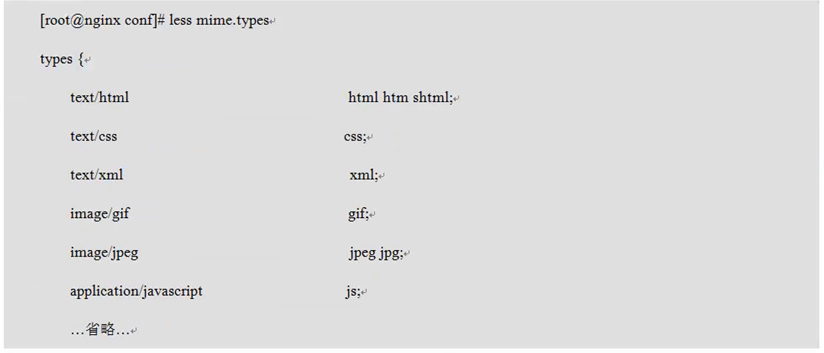
互联网的数据有很多种不同的数据类型，web服务器会把通过web传输的每个对象都打上名为MIME类型（MIME type）的数据格式标签。最初设计MIME（multipurpose Internet mail extension多用途因特网邮件扩展）是为了解决在不同的电子邮件系统之间搬移报文时存在的问题。MIME在电子邮件系统中工作得非常好，后来HTTP也支持了这个功能，用它来把数据描述并标记不同的数据内容类型。

当web服务器响应HTTP请求时，会为每一个HTTP对象数据加一个MIME类型。当web浏览器获取到服务器返回的对象时，回去查看相关的MIME类型，进行响应处理。

MIME类型存在于HTTP响应报文的响应头部信息里，它是一种文本标记，表示一种主要的对象类型和一个特定的子类型，中间由一条斜杠来分隔。下表为生产场景最常见的MIME类型：



可以从www中药服务器软件nginx的配置文件conf目录下，查看其支持的媒体（MIME）类型，命令及内容为：



**URL介绍**

URL，全称uniform resource location，中文翻译为中医资源定位符，也被称为网页地址（网址）。如果在网络上的门牌，是因特网上标准的资源的唯一地址。通俗地说，URL是Internet上用来描述信息资源的字符串，主要用在各种WWW客户端和服务器程序上。采用URL可以用一种同意的格式来描述各种信息资源，包括文件、服务器的地址和目录等。严格讲，每个URL都是一个URI，它标识一个互联网资源，并制定对其进行操作或取得该资源的方法。

**URL的格式**

URL的格式由下列三部分组成：

第一部分是协议，例：http：

第二部分是主机资源服务器IP地址或域名（端口号），例：[www.teitnatian.org](http://www.teitnatian.org)：

第三部分是主机资源的具体地址，如目录和文件名等，例：oldboy/index.html。

第一部分和第二部分之间用“://”符号隔开，第二部分和第三部分用“/”符号隔开。第一部分和第二部分是不可缺少的，第三部分可以省略。



**URI介绍**

URI，全称uniform resource identifier，中文翻译为同意资源标识符，是一个用于标识某一互联网资源名称的字符串。这个字符串在世界范围内唯一标识并定位某一个信息资源。互联网上每个可用的数据资源HTML、图片、视频等皆通过同意资源标识符进行定位。



**静态网页资源**

静态网页资源介绍

在网站设计中，纯粹HTML格式的网页（可以包含图片，视频、JS（前端功能实现），CSS（样式）等通常被称为“静态网页”，早期（2002年前）的网站大多都是由静态网页制作的，静态网页是相对于动态网页而言的，是指没有后台数据库、不含程序（如php、jsp、ASP）和可交互的网页。

静态网页资源的特定：

静态网页资源的特点是，开发者编写的是什么，它显示的就是什么、一旦编写完成，就不会有任何改变。静态网页的维护和更新相对比较麻烦，每个不同的网页都需要单独编辑更新，静态网页一般适用于更新较少的宣传展示型网站（例如：酒、家具、猪饲料宣传等）。是早期2002年以前多数中小网站展示的形式。

扩展名特征:静态网页资源的对应程序以及资源文件的常见扩展名为:

a)纯文本类程序或文件, hlm,hlm1,xm1,shlm1,js,css等。

b)图片类文件或数据文档, jpg,gifpng,bmp,txt,doc,ppt等。

c)视频类流媒体文件 mp4,swfavi,wmv,flv等。

静态网页资源有几个重要的特征特点:

(1)每个页面都有一个固定的uR]\_地:t止, 且uRl\_般以.hlm、 hlm1、 』hlm1等常见形式为后缀,而且地:t止中不含有问号“?”或“&”s等特殊符号。

(2)网页内容一经发布到网站服务器上, 无论是否有用户访问, 每个网页的内容都是保存在网站服务器文件系\_绩上的,也就是说,静态网页是实实在在保存在服务器上的文件实体,每个网页都是一个独立的文件。

(3)网页内容相对固定的,因此,容易被捜索引擎收录(容易被用户找到) (优点) 。

(4)网页没有数据库的支持,在网站制作和维护方面工作量较大,因此当网站信息量很大时完全依靠静态网页制作的方式比较困难(缺点) 。

(5)网页的交互性较差,在程序功能实现方面有较大的限制(缺点) 。

(6)网页程序在用户浏览器端解析,如 IE刘览器,这样程序解析效率更高, 由于服务端不进行解析,并且不需要读取数据库,因此可以接受更多的并发访问。当客户端向服务器请求数据时,服务器直接把数据从磁盘上返回(不做任何解析) ,当客户端拿到数据后,在浏览器端解析展现出来(优点) 。

顾客点餐🡪饭店给你准备原材料和工具，静态页面相当于让各科自己做，火锅店。

**静态网页语言：**

常见的静态网页语言有：HTML、js、CSS、XML、shtml等。

**静态网页核心特点：**

1、客户浏览器端解析程序，不需要读取数据库，性能和效率很高。

2、扩展名格式：a.纯文本类型程序或文件，htm,html,xml,shtml,js,css等。b.图片类文件或数据文档，jpg,gif,png,bmp,txt,doc,ppt等。c.视频类流媒体文件mp4,swf,avi,wmv,flv等。

3、后端没有数据库支持，和用户交互性差。

有关静态网页架构思想：

高并发高访问量的场景，做架构优化，关键的环节：就是把动态网页转成静态网页，不直接请求数据库和动态服务器，并且可以把静态内容推送到前段缓存中提供服务，可以提升用户体验，节约服务器和维护成本。

**动态网页资源**

动态网页资源介绍

所谓动态网页是与静态网页相对应而说的，也就是说，动态网页的URL后缀不是.htm、html、shtml、xml、js、css等静态网页的常见后缀扩展名形式，而是以.asp,aspx,hph,jsp,do,cgi等形式为后缀，并且一般在动态网页网址中经常会有标志性的符号---“？，&”，以及大多数情况后端需要数据库支持等。下图为动态网页地址的形式



**动态网页资源特点：**

1、网页扩展名：

常见的：asp,aspx,php,jsp,do,cgi等。

2、网页一般以数据库技术为基础，大大较低了网站维护的工作量。

3、采用动态网页技术的网站可以实现更多的功能，如用户注册、用户登录、在线调查、投票、用户管理、订单管理、发博文等等；

4、动态网页并不是独立存在于服务器上的一个个网页文件（php/jsp），当用户请求服务器上的动态程序时，服务器解析程序并读取数据库返回一个完整的网页内容。

5、动态网页中的“？”对搜索引擎的收率存在一定的问题，搜索引擎一般不会从一个网站的数据库中访问全部网页，或者出于技术方面的考虑，搜索蜘蛛一般不会去抓取网址中的“？”后面的内容，因此采用动态网页的网站在进行搜索引擎推广时需要做一定的技术处理（伪静态技术）才能适应搜索引擎的抓取的要求。

6、程序在服务器端解析，服务端如：🡪php引擎，Java容器（tomcat，resin，Jboss，weblogic），相当于顾客点餐，饭店里做饭做菜，耗时长，效率低，由于程序在服务端解析，因此，会消耗大量的CPU和内存等资源，并且多数还要读取数据库等服务，因此，访问效率远不如静态网页。

**动态网页资源核心特点：**

a.程序在服务端解析，加上后端需要数据库支持，因此访问效率较差。

b.URL地址后缀一般为asp,aspx,php,jsp,do,cgi，并且常伴随“?,&”等。

有关动态网页架构思想：

1、一般来说，静态网页的效率是动态网页的10-30倍。

2、动态网站效率很差，并发能力也很低，高并发场景中，应尽可能转换成静态网页提供服务。

3、动态转静态也要根据业务需求设计，例如，对于更新频繁的网站会产生数据不一致的情况，即用户看到的数据不会是最新的内容。

4、动态转静态，高并发网站必备的架构方案，是高级架构师的职责所在。

**伪静态网页**

伪静态就是通过某些技术（如rewrite）把动态网页的URL地址伪装成静态网页URL地址，但实质上用户访问的还是动态网页，只不过给用户看起来URL地址时静态的，因此，用户及某些搜索引擎就会误认为是静态网页。

那么，问题来了，为什么动态网页需要伪装成静态呢，为什么不能直接转成静态网页呢？

伪静态网页特点：

从网站的URL地址看，伪静态表面上看起来是静态内容（如地址结尾带HTML），这是通过rewrite规则来实现的URL地址重写。该写后的URL地址规范、美观，有利于搜索引擎抓取，以及用户访问体验加强。如：<http://oldboy.blog.51cto.com/2561410/803606，http://www.discuz.net/forum-3967-1.html>。这样的两个地址都是伪静态。因为伪静态网页还是动态网页，从性能上考虑，伪静态功能不但没有提升网站性能，反而会降低网站的性能。

**伪静态网页作用：**

由于搜索引擎无法 正确读取带参数的动态网页内容，造成网页中很多具有丰富信息的页面无法被搜索引擎收录，伪静态技术的作用是把动态网页URL改写成静态网页的URL，同样是消除了动态网页中的参数，但是并不需要生成任何网页，仅仅是改变了网页地址路径，这样做的主要目的有两个，一是让搜索引擎收录网站内容让更多的用户访问网站内容，而是已经提升用户访问体验，动态带的特殊符号等的URL看起来不如静态网页舒服。

**伪静态网页缺点：**

动态网页伪装静态网页，虽然可以让搜索引擎收录以及提升用户访问体验，但伪静态网页并不能提升网站的访问效率性能，从理论上说还会降低网站的性能，没有动态转静态网页效率高，但是对于一些并发不是超大的情况以及硬件资源充足的情况，伪静态还是一个不错的实用功能，因为，要把动态网页真正的转成静态网页，不是一朝一夕，需要技术团队通力配合才能行。

现在很多大型网站都采用了动态网页生成静态网页的技术，消除动态网页中的参数，是搜索引擎收录更多的内容，达到优化网站的效果，生成静态网页确实可以更好的解决搜索引擎收录的问题，并且可以提升网站的访问性能，因此，当网站遇到访问瓶颈时，并且有技术力量去做动态网页生成静态网页的架构设计，那就最好不过了，但是，有些更新频率很快的业务也不是不适合或者做动态转静态有很大难度，可以考虑设置伪静态。

**伪静态网页小结：**

1、利用rewrite技术将动态网页伪装成静态网页。

2、便于搜索引擎收录提升用户访问量以及用户体验。

3、访问性能没有提升，转换为静态还会消耗资源，因此性能反而会下降。

4、尽可能将动态网页转换成整整的静态页面。

5、并发量不是很大或者动态更新过于频繁的时候，用rewrite实现伪静态也是不错的。

6、伪静态网页的实现过程，一般是由运营产品提出需求，开发和运维共同实现的。

**生产web架构后话实战方案**

由于静态网页程序在客户端解析，大大降低了服务器端的访问压力，因此解析效率更高，在实际高并发网站架构中，我们可以考虑把用户请求的数据解析后存成静态文件防御磁盘中或放于内存中，来降低动态服务器的压力，节约企业成本，提升用户体验。有关高并发网站架构从动态转静态的内容，可以参考老师的文章：

浅谈千万级PV/IP规模高性能高并发网站架构<http://oldboy.blog.51cto.com/256410/736710>.

**高并发网站架构从动态转静态不用业务产品实现的例子：**

* 门户新闻业务

新闻网站的特点是一旦发布完成，几乎不会再改动网页内容。因此，对于新闻的业务内容静态化相对比较简单：

第一步：程序支持发布内容直接转成静态。

第二步：运营编辑人员发布新闻网页后，后台程序立刻将动态网页生成静态文件。

第三步：运维人员通过发布触发把生成的静态网页发布到实现搭建好的公司缓存集群服务器上，或者把静态内容同步到购买的全国所有CDN服务器节点上，然后，在提供给用户提供访问浏览。

* 视频网站业务（一旦发布完成，几乎不会改动）

优酷视频网，用户上传视频，需要经历审核-转码的过程，大概1-2个小时，审核-转码—推送（同步到全国所有CDN服务器节点）。

blog/bbs/sns/微博社区业务/电商（淘宝，京东）（用户发布完成，可能会随时更新）

异步方式，消息中间件、消息队列。

**网站流量度量术语**

IP-Internet protocol

IP（独立IP）：即Internet protocol，这里指独立IP数，独立IP数是指不同IP地址的计算机访问网站的被计算的总次数，独立IP数是衡量网站流量的一个重要指标，一般一天内（0:00-24:00）内相同IP地址的客户端访问网站页面只被计算为一次，记录独立IP的时间可为一天或一个月。目前通用的标准一般为“一天”。

在做的各位在教室内局域网中同时打开博客（http://oldboy.blog.51cto.com），请问对于51CTO网站是几个独立IP？答：是一个独立IP，这是因为，国内几乎所有的公司尚未都是局域网共享上网，即通过路由器NAT地址转换上网，每个计算机在局域网内的私有IP是不同的，但是大家出外网上网，就必须都要被路由器吧每个私网地址转换成了路由器接口的公网IP了，所以，对于往后在哪来说一天内多个相同IP的客户端访问计算为一个独立IP。

请问一个客户机用户是通过ADSL等直接拨号上网，但是上网的时候偶尔掉线，一共重新拨号了3次，然后每次都继续打开老师的博客地址，请问此时，网站独立IP数是多少？答：是3个独立IP。

由此可见，通过独立IP数度量网站访问量，和实际访问情况不是很匹配。国内的企业、学校等上网多数是NAT上网的，一个独立IP背后可能有数十上百个客户端访问， 独立IP数虽然不是很准确，但却是IP技术人员比较关心的一个衡量网站的指标。

PV-page view

PV（访问量）：即page view，中文翻译即页面浏览，即页面浏览量或点击量，不管客户端是不是相同，也不管IP是不是相同，用户每次访问一个网站页面都会被计算一个PV。

PV的具体度量方法就是从客户浏览器发出一个队web服务器的请求（request），web服务器街道这个请求后，将该请求对应的一个网页（page）发送给浏览器，就产生了一个PV。这里有一个问题，就是只要这个请求发送给了浏览器，无论这个页面是否完全打开（或下载完成），那么都是会被计数为1个PV（服务器日志），一般为了防止用户快速刷PV，很多网站把PV的统计程序放在页面的最下面。

用PV恒联网站，PV数反映的是浏览某网站的页面数量，每刷新一次页面也算一次。因此，可以说PV数与来访用户的数量成正比，但PV数并不是页面的来访者数量，而是网站被访问的页面数量，因为一个来访者可能产生多个PV。

问：如果一个用户访问赶集网或58同城，你觉得可能会产生多少PV？解答：平均可能会10几个-几十个PV，一个来访者访问网站的PV数多少是和网站提供的业务直接相关的，例如：这些分类网站，用户浏览网站，可能是为了找房子，找工作，因此一个用户访问的页面会很多，因此PV就会很多。

PV（page view）是网站被访问的页面数量的一个指标，但不能直接知道有多少人访问了这个网站。

一个来访者访问网站，PV数可能产生若干，但是独立IP数就是为1个，因此，如果对比一个网站的独立IP数和PV数，不难看出，PV数一定会大于独立IP数的，视网站的业务而定，例如这种分类门户可能要达到10:1，甚至更多。

UV- （unique visitor）

UV（独立访客）：即unique visitor，同一台客户端（PC或移动端）访问网站被计算为一个访客。一天（0:00-24:00）内相同的客户端访问同一个网站值被计算一次UV，UV（独立访客）一般是以客户端cookie等技术作为统计依据的，实际统计会有误差。

考虑到一台客户端电脑可能会有多人使用情况，因此，UV实际上并不一定时独立的自然人访问。 （理解：IP,PV,UV的概念）

**企业网站对PV/IP/UV的度量**

IP的度量：

1.分析所有web服务器的访问日志信息，对IP地址段去重后计数，这是IP人员的基本计算手段。

2.在网站的每一个（所有）页面结尾，嵌入js等统计程序代码，当用户加载网页后，IP即传给统计IP的服务器，这种方法一般被第三方统计公司或者企业内部开发日志分析程序时使用。

3.用第三方大家比较信任的统计工具如：谷歌的统计（GA）。

IP的统计方法简单、易用，因此，成为了多数网站衡量网站流量的重要指标之一。

PV的度量：

1.分析web服务的访问日志（需要排除,js,css，各种图片的日志信息），值计算HTML，PHP等页面URI。

2.在网站的每一个（所有）页面结尾，嵌入js等统计程序代码，当用户加载网页后，IP即传给统计IP的服务器，这种方法一般被第三方统计公司或者企业内部开发日志分析程序时使用。

3.用第三方大家比较信任的统计工具如：谷歌的统计（GA）。

PV的统计方法也简单、易用，因此，成为了多数网站衡量网站流量的重要指标之一。

提示：IP和PV也是Linux运维人员要掌握的重点。

面试题：统计往后在哪某一时间段所有IP的对应的PV数排序，前10.

UV的度量：

1.通过客户端HTTP请求报文分析。

一个客户端会多次请求网站服务器，每次HTTP请求都会携带客户端自身的大量信息，比如：IP地址，请求发出的时间，浏览器版本，操作系统版本等。网站服务器对这些请求进行分析，如果这些请求满足一些共同特征，比如来自同一个IP地址，且浏览器版本，操作系统版本相同，请求时间又相近等，如果满足这一系列的定义，那么久可以认为这些请求时来自同一个客户端，那么多个页面访问也只算一个UV。共同特征的定义是由服务器放决定的。通常，用IP地址+其他特征共同定义的情况较多。此种度量方法无法解决一下问题，例如：1.多个人的电脑的软硬件经常雷同，并且是一个公司或者学校的人。2.多个人公用一个电脑的情况。

2.通过cookie鉴别

当客户端第一次访问某个网站服务器的时候，网站服务器会给这个客户端的电脑发出一个cookie，通常放在这个客户端电脑的C盘当中，在这个cookie中会分配一个独一无二的编号，这其中会记录一些访问服务器的信息，如访问时间，访问了哪些页面等等。当你下次再访问这个服务器的时候，服务器就可以直接从你的电脑中找到上一次放进去的cookie文件，并且对其进行一些更新，但那个独一无二的编号时不会变的。如果在一定时间内，服务器发现2个来访者对应的是一个编号，那么自然可以认为这个来源于同一个来访者了。就计算1个UV。

使用cookie的方法要比分析客户端http请求头部信息分析更精确些。但也存在一些问题，比如：有的客户端为了保证更高级别的安全，关闭了cookie的功能；或者是有些客户端设置了在退出页面时自动删除cookie，亦或你经常自己去手动删除cookie，那么这个方法就不那么精确了。

因此，以上两个方法都只能得到近似的UV，而不是绝对精确的。

UV的度量相对IP和PV来说，不但方法麻烦，而且要开发比较复杂的程序系统计算才能得到期望的结果，因此，Linux运维领域，大家涉及的较少，一般企业市场及运行人员可能会关注网站的UV。

IP/PV/UV的区别

通过一个访问例子来讲解

例如某城市的一个网吧里，有10个人都进入了[www.etiantian.org的网站，假设每个人平均访问了5](http://www.etiantian.org的网站，假设每个人平均访问了5)个页面，但是这个网吧对外出口是一个公网IP（注意：也可以配置多个IP出口，此处不计特殊情况），所以，对于etiantian网站来说，只会计算一个独立IP访问，但是因为网吧里有10个人在访问该网站，并且平均 都访问了5此，因此，对于etiantian网站来说，PV数量就是10\*5=50个PV，而因为有10个人访问，就是10个不同的客户端访问，因此UV为10.

即上述例子中：网站独立IP数1个，PV数50个，UV数10个。

通过上述结果，可得出一个结论，一个网站的独立IP数量要比网站实际访问的PV数量小得多，通常情况下（国内互联网环境），网站的UV数也会大于独立IP数。

PV数高说明访问的页面数多，但是不一定就代表来访者多；但PV数一定与来访者的数量成正比，但是PV并不直接决定页面的真实来访者数量。比如一个网站只有你一个人进来，铜鼓哦不断的刷新页面，也可以制造出非常高的PV。PV数多，用户访问网站页面的总数量多，通常服务器的压力会大一些。

**并发连接**

在面试过程中Linux运维人员经常会被问到：你的公司网站最大并发是多少这样的问题。

那么到底什么是并发？怎么理解并发呢？

虽然查阅整个互联网，但还是没有查到让人信服的确切说法：下面是网上的一些说法

A种理解：网站服务器每秒能够接受的最大用户请求数。

B种理解：网站服务器每秒能够响应的最大用户请求数。

C种理解：网站服务器在单位时间内能够处理的最大连接数。

虽然A,B的理解占据了很多IT人员的大多数，但是，按照老师的理解，C中理解更为准确一些。

举个单位时间段的例子说明：

去餐馆吃饭：餐馆里一共有10张桌子，每桌最多坐4个人同时吃饭，那么一般人的理解，这个餐馆能够接收的吃饭的并发人数为10\*4，即40个并发，这里就没有考虑时间问题，1秒并发可以是40个，10分钟内并发可能也是40个。因为这里还有一个因素，就是每个人吃饭时长的问题，如果平均每个人10分钟吃完，那么可以说10分钟内，这个餐馆的并发为40个，而不是每秒钟并发40个，因为，第一秒可以是40个人同时进来，但是第二秒就无人可进了（满员了），如果说10分钟并发是40个，下一个10分钟还能是40个，第三个10分钟还可以是40个。这就是往后在哪服务器在单位时间内能够处理的最大连接数。

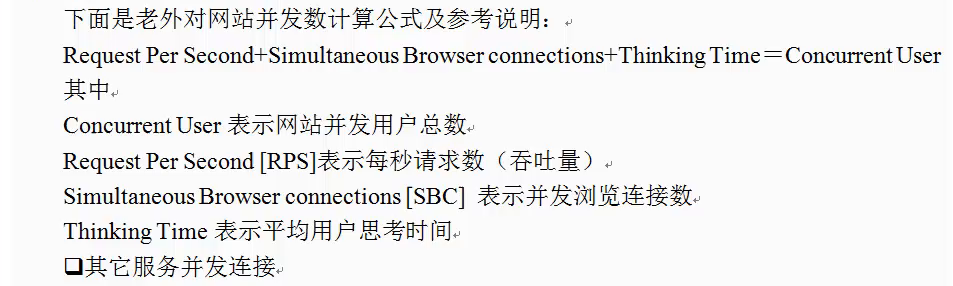
再举个同一个时刻例子说明：

高速公路每个方向都有2条车道，那么，同一时刻并发的车辆为两辆，并且并发可以永远为2，如果按秒计算，每秒的并发可能就可以有10几辆，这个例子和餐馆不同，因为高速路处理并发不需要处理时间，但是对于web服务器来讲，需要花费时间处理请求，这个请求可能是1秒或数秒，因此说，并发不应该知识访问的请求数，而是服务器同时处理的并发数，并且单位时间不一定是秒，可能是一个链接的处理周期内的连接数。

对于网站服务器来说，所谓的并发就是单位时间内，服务器能够同时处理的最大连接数（或请求数），因为有的请求1秒结束，有的请求可能10秒才结束（业务程序及配置不同），因此，网站并发不是客户端每秒的并发请求数，而是服务器在一段时间内（1秒或者数秒内）可以处理的最大连接数，这个链接即包含正在建立的连接，也包含已经建立的连接。

例如：某个网站的并发是5000

意味着：单位时间内（理解为1秒或数秒内），正在处理的连接数，正在建立的连接数，加起来一共是5000个。



1、QPS每秒查询率（query per second）

每秒查询QPS是对一个特定的查询服务器在规定时间内锁处理流量多少的衡量标准，运维工作中，DNS系统以及数据库等服务的查询性能经常用每秒查询率来衡量。

2、IOPS（input/output operations second），即每秒进行读写（I/O）操作的次数，多用于数据库等场合，衡量随机访问的性能。存储端的IOPS性能和主机端的IO是不同的，IOPS是指存储每秒可接受多少次主机发出的访问，主机的一次IO需要多次访问存储才可以完成。例如，主机写入一个最小的数据块，也要经过“发送写入请求、写入数据、收到写入确认”等三个步骤，也就是3个存储端访问。

常见企业网站排名及PV/IP访问量



提示：以上数据来自第三方<http://alexa.chinaz.com/alexa_more.aspx>网站数据，仅供参考，不同的统计程序差别也很大，有一定误差，实际访问量要比此表答，具体的误差和chinaz.com的统计方法有关，后面的最大并发以及机器数量级别为作者根据访问量级业务类型估算而来，对初学者是一个参考。

**当前互联网主流web服务说明**

当前互联网主流web服务软件有：

1、Apache 🡨这是中小型web服务的主流，web服务器中的老大哥。

2、nginx 🡨大型往后在哪web服务主流，曾经服务器中的初生牛犊，现已长大。

nginx的分支tengine（http://tengine.taobao.org/）目前也在飞速发展。

3、lighttpd 🡨这是一个不温不火的优秀的web软件，社区不活跃，静态解析效率很高。在nginx流行前，大并发静态业务的首选，国内百度贴吧，豆瓣等众多网站都用过Lighttpd。

常用来提供动态服务的软件：

1、php（fastcgi）大中小型往后在哪都会使用，动态网页语言PHP程序的解析容器。

a.配合Apache解析动态程序，这里的PHP不是fastcgi守护进程模式，而是mod\_php5.so(module)

b.配合nginx解析动态程序，这里的PHP常用fastcgi守护进程模式提供服务。

2、tomcat 🡨中小企业动态web服务主流，互联网Java容器主流（JSP，do）

3、resin 🡨大型动态web服务主流，互联网Java容器主流（JSP，do）。

4、IIS（Internet information service）🡨微软的Windows下的web服务软件（ASP，aspx）

静态程序服务软件Apache

Apache软件有几个重要的版本系列，分别为：1.3、2.0、2.2、2.4系列，其中，1.3和2.0系列已经成为过去式了，官方网站也看不见其踪影了，目前主流的Apache为2.2系列，正在向2.4系列过度阶段，建议使用2.2系列。

官网地址：<http://www.apache.org/>

静态服务软件nginx

nginx的版本只有一个系列，但是版本更新很快，仅仅半年就有数个版本，这也看出来社区的活跃程度，具体内容参见官方文档地址：<http://www.nginx.org/>以及<http://nginx.org/en/docs/，目前最新稳定版为1.6.2>

动态服务软件resin

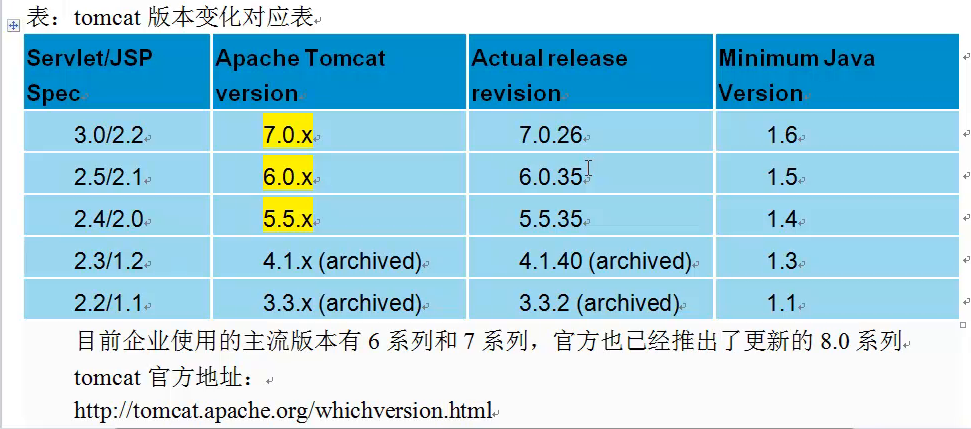
resin官方号称是世界上最快的web服务，是大型动态web服务主流，互联网Java程序的解析器容器，百度，人人都曾使用过resin。

其发展的主要版本：3.0、3.1、4.0，目前企业较多的使用3.1系列，并正在向4.0过渡。



动态服务软件tomcat

tomcat一直是中小企业动态web服务的主流，常用来作为解析Java程序，其版本发展变化如下表：



动态服务器软件PHP

PHP软件是大中小型网站程序前台页面开发的首选，存世开源软件众多，也是中小企业网站开发的首选，它是动态网页语言PHP程序的解析容器。PHP的版本系列有5.2、5.3、5.4、5.5、5.6，其中最经典的版本为5.2系列，企业应用的主流版本可以说是百花争艳。

指的注意的是PHP提供解析的方式，配合Apache解析动态程序，是mod\_php5.so（module）模块的方式，配置nginx解析动态程序，常用fastcgi守护金恒模式提供服务。

官网：<http://php.net/>

章节重点回顾：

用户访问网站的基本流程

NDS系统的解析原理

HTTP协议通信原理

HTTP协议、请求报文、响应报文、状态码

动、静态概念特点以及伪静态技术

动态转静态web优化方案

IP、PV、UV的概念和区别

了解常用WWW服务软件特点

Apache、nginx、PHP（fastcgi）、tomcat、resin

本章知识相关面试考试题：

1、请描述NDS系统的解析原理？🡨运维人员必会。

2、请描述http协议的原理？🡨运维人员必会。

3、请问你的公司的网站访问量是多少？🡨运维人员必会。

# 2、nginx服务

**1、http协议包含很多功能：**

上网的www是http功能之一，www服务默认端口80，OSI第7层，应用层协议。https 默认端口443，加密的http服务。

**2、实现WWW服务的常用web软件：**

产品包括：nginx，Apache（静态web软件）

**3、经典流行的web组合**

lamp（Linux Apache MySQL PHP）===🡺经典

lnmp（Linux nginx MySQL PHP）===🡺国内流行

nginx：（engine x）

比较早的，比较经典的web服务组合

lamp（Linux Apache MySQL PHP）

近3年的一个新的web服务组合

lnmp（Linux Apache MySQL PHP）

lemp（Linux （engine x） MySQL PHP）

**4、nginx介绍**

nginx是www服务软件，俄罗斯人开发，是一款开源的软件，性能很高。开源的www服务软件一共780K。

nginx本身是一款静态（HTML，js，CSS，JPG等）www软件，不能解析动态的PHP，JSP，do文件。若要解析此类文件的话，需要安装PHP引擎，JSP与do需要安装tomcat。

最优的特点：

1、静态小文件（低于1M情况下）支持高并发，同时占用的资源很少，3W并发，10个进程，内存150M。

nginx服务端支持的平台：Unix，Linux，Windows都可以。

nginx排名：第二位，并且上升趋势。在全球25%使用率。

<http://w3techs.com/technologies/overview/web> server/all

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

国内网站使用占比，nginx更多一些。

**5、nginx服务从大的方面的功能：**

a.【www web服务】最为核心的功能，邮件服务，邮件代理。

b.负载均衡（反向代理proxy）

c．web cache（web缓存），相当于squid（CDN主要使用suid）。

**6、nginx特点：**

1、配置简单，灵活，轻量。

2、占用资源少，2W并发，开10个线程服务，内存消耗几百M。

3、功能种类比较多（web，cache，proxy），每一个功能都不是特别强。

4、支持epoll模型。使得nginx可以支持高并发！Apache使用的是select模型。

5、nginx可以配合动态服务（fastcgi接口）

6、利用nginx可以对IP限速，可以限制连接数。

它所具备的其他WWW服务特性如下：

支持基于名字、端口以及IP的多虚拟主机站点；

支持rewrite模块，支持URI重写正则表达式匹配；

支持基于客户端IP地址和HTTP基本认证的访问控制；

支持http响应速率限制；

支持同一IP地址的并发连接或请求数限制

**7、nginx的应用场合：**

1、提供静态服务器（图片，视频服务），另一个Lighttpd。并发：几万并发。HTML，js，CSS，flv，JPG，GIF等。类似Lighttpd。

2、提供动态服务，nginx+fastcgi的方式运行PHP，JSP。动态并发：500-1500 Apache+PHP，Lighttpd+fcgi PHP

3、提供反向代理（proxy）服务，或称为负载均衡。日PV2000W以下，并发1万以下，都可以直接用nginx做反向代理。paproxy（软件），F5、A10（硬件）

4、提供缓存服务，类似squid，varnish，ATS。

**8、nginx主要应用场景：**

1、web服务器（首选）静态文件，nginx首选。动态文件，配合fastcgi支持PHP

2、反向代理（负载均衡），1000-2000W PV，并发6000.

3、缓存（不是很多）类似squid，varnish

4、提供动态服务，nginx+fastcgi的方式运行PHP，JSP，动态并发：500-1500

nginx通过fastcgi客户端和fastcgi服务端（PHP）通信。 Apache+PHP，Lighttpd+fcgi php

**9、nginx支持虚拟主机：**

一个server标签段就是一个虚拟主机。

1、基于域名的虚拟主机。通过域名来区分虚拟主机 🡺应用：外部网站。\*\*\*\*\*

2、基于端口的虚拟主机。通过端口来区分虚拟主机 🡺应用：公司内部网站，网站的后台。\*\*\*

3、基于IP的虚拟主机。几乎不用。不支持ifconfig别名，配置文件可以。

10、nginx安装：

<http://nginx.org>

1、安装pcre

介绍：

pcre全称（Perl compatible regular expressions），中文Perl兼容正则表达式，官方站点为：<http://www.prcr.org/>

HTTP rewrite module require the pcre library.

两种安装方法：

法1：

wget <http://sourceforge.net/projects/pcre/files/pcre/8.30/pcre-8.30.tar.gz>

tar zxf pcre-8.30.tar.gz

cd pcre-8.30/

. /configure

mkde && mkde install

cd ..

法2：

rpm –qa pcre pcre-devel

yum install –y pcre pcre-devel

rpm –qa pcre pcre-devel

b.SSL modules require the openssl library

yum install openssl-devel –y

c.安装nginx

useradd nginx –s /sbin/nologin -M

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

tar zxf nginx-1.6.3.tar.gz

cd nginx-1.6.3

./configure –prefix=/application/nginx-1.6.3

--user=nginx –group=nginx –with-http\_ssl\_module

-–with-http\_stub\_status\_module

make && make install

cd ..

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

操作命令如下：

yum安装pcre（老师推荐的安装方法）

提示：禁止使用yum remove卸载应用！！！！！

[root@web01 ~]# cat /etc/redhat-release

CentOS release 6.7 (Final)

[root@web01 ~]# uname -r

2.6.32-573.el6.x86\_64

[root@web01 ~]# uname -m

x86\_64

[root@web01 ~]# rpm -qa pcre pcre-devel ####--开始安装pcre

[root@web01 ~]# yum install openssl-devel –y ####--开始安装SSH相关服务

[root@web01 ~]# mkdir -p /home/oldboy/tools ###首先创建目录

[root@web01 ~]# cd /home/oldboy/tools

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

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

nginx-1.6.3.tar.gz

[root@web01 tools]# ls nginx-1.6.3.tar.gz -lk

-rw-r--r-- 1 root root 787 Apr 8 2015 nginx-1.6.3.tar.gz

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

nginx-1.6.3/

nginx-1.6.3/auto/

[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]#

[root@web01 nginx-1.6.3]# tree|wc –l ###一共365个文件。

404

[root@web01 nginx-1.6.3]# ./configure –help ###编译查看里面的参数模块。

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

[root@web01 nginx-1.6.3]# echo $? ####查看安装是否完成。

0

root@web01 nginx-1.6.3]# make

[root@web01 nginx-1.6.3]# echo $?

0

[root@web01 nginx-1.6.3]# make install

[root@web01 nginx-1.6.3]# echo $?

0

[root@web01 nginx-1.6.3]# useradd nginx -s /sbin/nologin –M ###创建一个用户

[root@web01 nginx-1.6.3]# id nginx

uid=506(nginx) gid=506(nginx) groups=506(nginx)

[root@web01 nginx-1.6.3]# cd ..

[root@web01 tools]# ll -d /application/nginx-1.6.3/ ###创建软链接

drwxr-xr-x 6 root root 4096 Nov 16 16:51 /application/nginx-1.6.3/

[root@web01 tools]# ln -s /application/nginx-1.6.3/ /application/nginx

[root@web01 tools]# /application/nginx/sbin/nginx ###启动服务

[root@web01 tools]# ps -ef|grep nginx|grep -v grep ###查看服务是否开启

root 8424 1 0 17:04 ? 00:00:00 nginx: master process /application/nginx/sbin/nginx ##主进程

nginx 8425 8424 0 17:04 ? 00:00:00 nginx: worker process

[root@web01 tools]# ss -lntup|grep nginx ###查看端口号

tcp LISTEN 0 511 \*:80 \*:\* users:(("nginx",8424,6),("nginx",8425,6))

[root@web01 tools]# curl 127.0.0.1 ###检查是否安装成功。

<!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>

[root@web01 tools]# ####此时在浏览器输入设备的IP地址，即可浏览首页





**如何查看nginx编译时的参数。**

可采用如下命令查看：

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

nginx version: nginx/1.6.3

built by gcc 4.4.7 20120313 (Red Hat 4.4.7-16) (GCC)

TLS SNI support enabled

configure arguments: --prefix=/application/nginx-1.6.3 --user=nginx --group=nginx --with-http\_ssl\_module --with-http\_stub\_status\_module

[root@web01 tools]#

**排错命令：**

[root@web01 tools]# cat /application/nginx/logs/error.log

2015/11/16 17:08:32 [error] 8425#0: \*3 open() "/application/nginx-1.6.3/html/favicon.ico" failed (2: No such file or directory), client: 10.0.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "10.0.0.8"

**小试牛刀：**

[root@web01 tools]# cd /application/nginx

[root@web01 nginx]# ll

total 36

drwx------ 2 nginx root 4096 Nov 16 17:04 client\_body\_temp ####带temp的都是临时文件

drwxr-xr-x 2 root root 4096 Nov 16 16:51 conf

drwx------ 2 nginx root 4096 Nov 16 17:04 fastcgi\_temp

drwxr-xr-x 2 root root 4096 Nov 16 16:51 html

drwxr-xr-x 2 root root 4096 Nov 16 17:04 logs

drwx------ 2 nginx root 4096 Nov 16 17:04 proxy\_temp

drwxr-xr-x 2 root root 4096 Nov 16 16:51 sbin

drwx------ 2 nginx root 4096 Nov 16 17:04 scgi\_temp

drwx------ 2 nginx root 4096 Nov 16 17:04 uwsgi\_temp

[root@web01 nginx]# ls -l|grep -v temp

total 36

drwxr-xr-x 2 root root 4096 Nov 16 16:51 conf ###配置文件的目录

drwxr-xr-x 2 root root 4096 Nov 16 16:51 html ###站点目录（默认网站目录）pid

drwxr-xr-x 2 root root 4096 Nov 16 17:04 logs ###访问、错误日志

drwxr-xr-x 2 root root 4096 Nov 16 16:51 sbin ###启动命令

配置一个默认的站点

[root@web01 nginx]# cd html/

[root@web01 html]# ls

50x.html index.html

[root@web01 html]# vi index.html ####将配置文件内容更改为下列：

<head><title>oldboy,s Nginx server blog.</title></head> <body> Hi,I am oldboy. My blog add

ress is

<a href="http://oldboy.blog.51cto.com">http://oldboy.blog.51cto.com</a>

</body>

</html>

"index.html" 4L, 187C written

[root@web01 html]# ls

50x.html index.html



[root@web01 html]# cd ..

[root@web01 nginx]# cd conf/

[root@web01 conf]# ls -l nginx.conf

-rw-r--r-- 1 root root 2656 Nov 16 16:51 nginx.conf

[root@web01 conf]# grep root nginx.conf

root html;

root html;

# root html;

# deny access to .htaccess files, if Apache's document root

# root html;

# root html;

[root@web01 conf]# lsof -i :80

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

nginx 8424 root 6u IPv4 26795 0t0 TCP \*:http (LISTEN)

nginx 8425 nginx 6u IPv4 26795 0t0 TCP \*:http (LISTEN)

**nginx软件功能模块说明**

**nginx软件之所以强大，是因为它具有众多的功能模块，下面列出了企业常用的重要模块。**

**分为两大块：**

nginx的核心模块（core functionality）

nginx核心功能模块负责nginx的全局应用，主要对应朱配置文件的main区块和evetns区块区域，这里有很多nginx必须的全局参数配置。有关核心功能模块的详细信息官方地址：http：//nginx.org/en/docs/ngx\_core\_module.html

标准的http功能模块集合

这些标准的http功能模块，虽然不是nginx软件所必须的，但是都是很常用的，因此绝大部分默认情况都会自动安装到nginx软件中，不建议擅自改动，保留软件的默认配置即可。

表5-1为企业场景常用的nginx http功能模块汇总

|  |  |
| --- | --- |
| **Nginx http功能模块** | **模块说明** |
| ngx\_http\_core\_module | 包括一些核心的http参数配置，对应nginx的配置为http区块部分。 |
| ngx\_http\_access\_module | 访问控制模块，用来控制网站用户对nginx的访问。 |
| ngx\_http\_gzip\_module | 压缩模块，对nginx返回的数据压缩，属于性能优化模块。 |
| ngx\_http\_fastcgi\_module | Fastcgi模块，和动态应用相关的模块，例如PHP。 |
| ngx\_http\_proxy\_module | Proxy代理模块。 |
| ngx\_http\_upstream\_module | 负载均衡模块，可以实现网站的负载均衡功能及节点的健康检查。 |
| ngx\_http\_rewrite\_module | URL地址重写模块。 |
| ngx\_http\_limit\_conn\_module | 限制用户并发连接数及请求数模块。 |
| ngx\_http\_limit\_req\_module | 限制nginx request processing rate 根据定义的key。 |
| ngx\_http\_log\_module | 访问日志模块，以指定的格式记录nginx客户访问日志等信息。 |
| ngx\_http\_auth\_basic\_module | Web认证模块，设置web用户通过账号密码访问nginx。 |
| ngx\_http\_ssl\_module | SSL模块，用于加密的http连接，如HTTPS。 |
| ngx\_http\_stub\_status\_module | 记录nginx基本访问状态信息等的模块。 |

在生产环境中，配置、调整及优化nginx软件，主要就是根据这些模块的功能修改相应的参数来实现的。通过官方地址<http://nginx.org/en/docs/>可以查看到上述及更多模块的详细使用帮助。

**nginx的目录结构说明**

为了更了解nginx软件的全貌，有必要介绍下nginx安装后整体的目录结构及文件功能。可通过执行tree /application/nginx/命令来查看，其内容如下：

[root@web01 nginx]# tree

.

├── client\_body\_temp

├── conf ##这是nginx的所有的配置文件的目录，及其重要。

│   ├── fastcgi.conf ##fastcgi相关参数的配置文件。

│   ├── fastcgi.conf.default ##fastcgi.conf的原始备份。

│   ├── fastcgi\_params ##fastcgi的参数文件。

│   ├── fastcgi\_params.default

│   ├── koi-utf

│   ├── koi-win

│   ├── mime.types ##媒体类型。

│   ├── mime.types.default

│   ├── nginx.conf ##这是nginx默认的主配置文件

│   ├── nginx.conf.default

│   ├── scgi\_params ##scgi相关参数文件，一般用不到。

│   ├── scgi\_params.default

│   ├── uwsgi\_params ##uwsgi相关参数文件，一般用不到。

│   ├── uwsgi\_params.default

│   └── win-utf

├── fastcgi\_temp ##fastcgi临时数据目录

├── html ##这是编译安装时nginx的默认站点目录，类似Apache的默认站点htdocs目录。

│   ├── 50x.html ##错误页面优雅替代显示文件，例如：出现502错误时会调用此页面。

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

│   └── index.html ##默认的首页文件，在实际环境中，我们习惯用（注意字眼不是必须）index.html、index.php、index.jsp来做网站的首页文件。首页文件名字是在nginx.conf中事先定义好的，具体参数为：index、index.html、index.htm注意与Apache参数不同directory index index.html

├── logs ##这是nginx默认的日志路径，包括错误日志及访问日志。

│   ├── access.log #这是nginx的默认访问日志文件，使用tail –f access.log可以时时观看用户访问情况信息。

│   ├── error.log #这是nginx的错误日志文件，如果nginx出现启动故障等问题，一定要看看这个错误日志

│   └── nginx.pid #nginx的pid文件，nginx进程启动后，会把所有进程的ID号写到此文件。

├── proxy\_temp #临时目录

├── sbin #这是nginx命令的目录，如nginx的启动命令nginx。

│   └── nginx #nginx的启动命令nginx

├── scgi\_temp #临时目录

└── uwsgi\_temp #临时目录

9 directories, 21 files

提示：未做注释的目录，为生产环境中比较少用的目录，因此建议不要理会。

**nginx主配置文件nginx.conf**

nginx主配置文件nginx.conf是一个纯文本类型的文件（其他配置文件也是如此），它位于nginx安装目录的conf目录下，整个配置文件是以区块的形式组织的。一般，每个区块以一个大括号“{}”来表示，区块可以分为几个层次，整个配置文件中main区位于最高层，在main区下面可以有events区、http区等层级，在http区又包含有一个或多个server区，每个server区又可有一个或多个location区，整个nginx配置文件nginx.conf的主体框架为：

[root@web01 conf]# less nginx.conf

1 worker\_processes 1;

2 #error\_log logs/error.log; 1-3行Main区，nginx核心功能模块

3 #pid logs/nginx.pid;

4

5 events {

6 worker\_connections 1024; 5-7行events区，nginx核心功能模块

7 }

8

9 http { 3行http区开始，nginx http核心模块

10 include mime.types;

11 default\_type application/octet-stream;

12 sendfile on;

13 keepalive\_timeout 65;

14 server {

15 listen 80;

16 server\_name localhost

17

18 location / { 18-21行location区块

19 root html;

20 index index.html index.htm;

21 }

22 location = /50x.html {

23 root html;

24 }

25 }

26

27 #server {

28 # listen 80;

29 # server\_name somename alias another.alias;

30

31 # location / {

32 # root html;

33 # index index.html index.htm;

34 # }

35 # location / 50x.html{

36 # root html;

37 # }

38 }

39 } 39行http区块结束。

**基于域名的虚拟主机配置实战**

nginx配置文件有数百行，可以先去掉所有的默认注释行，来最小化理解nginx的核心配置参数。

[root@web01 conf]# egrep -v "#|^$" nginx.conf.default #去掉包含#和空格的内容

worker\_processes 1; ###🡨worker进程的数量

events { ###🡨事件区块开始

worker\_connections 1024; ###🡨每个worker进程支持的最大连接数

} ###🡨时间区块结束

http { ###🡨http区块开始

include mime.types; ###🡨nginx支持的媒体类型库文件包含。

default\_type application/octet-stream; ###🡨默认的媒体类型

sendfile on; ###🡨开启高效传输模式

keepalive\_timeout 65; ###🡨连接超时

server { ###🡨第一个server区块开始，表示一个独立的虚拟主机站点。

listen 80; ###🡨提供服务的端口，默认80

server\_name localhost; ###🡨提供服务的域名主机名。

location / { ###🡨第一个location区块开始。

root html; ###🡨站点的根目录，相对于nginx安装目录。

index index.html index.htm; ###🡨默认的首页文件，多个用空格分开。

} ###🡨第一个location区块结果。

error\_page 500 502 503 504 /50x.html; ###🡨出现对应的http状态码使用50x.html回应。

location = /50x.html { ###🡨location区块开始，访问50x.html.

root html; ###🡨指定对应的站点目录为html。

}

}

} ###🡨http区块结束。

"nginx.conf" 22L, 484C 1,1 All

[root@web01 conf]# egrep -v "#|^$" nginx.conf.default >nginx.conf #将文件被注释的内容放到nginx.conf文件中（安全最小化）。

[root@web01 conf]# vim nginx.conf ##开始编译配置文件，删除了最后三行，又创建了两个虚拟主机，具体内容如下

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/www;

index index.html index.htm;

}

}

server {

listen 80;

server\_name bbs.etiantian.org;

location / {

root html/bbs;

index index.html index.htm;

}

}

}

~

"nginx.conf" 26L, 566C

[root@web01 conf]# mkdir -p ../html/{www,bbs} ###由于配置文件中的www和bbs没有目录，顾创建两个。

[root@web01 conf]# tree ../html/

../html/

├── 50x.html

├── bbs

├── index.html

└── www

2 directories, 2 files

[root@web01 conf]# echo "www.etiantian.org" > ../html/www/index.html

[root@web01 conf]# echo "bbs.etiantian.org" > ../html/bbs/index.html

[root@web01 conf]# cat ../html/{www,bbs}/index.html

www.etiantian.org

bbs.etiantian.org

[root@web01 conf]# /application/nginx/sbin/nginx –t ###检查语法，当出现下面两行提示，表示操作正常。

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]# /application/nginx/sbin/nginx -s reload ###加载使其生效

[root@web01 conf]# vim /etc/hosts ###做一个NDS将域名与IP放入配置文件中

10.0.0.8 www.etiantian.org bbs.etiantian.org

[root@web01 conf]# ping [www.etiantian.org](http://www.etiantian.org) ###ping一下两个域名，发现返回的IP就是服务器的IP，表示正常。

PING www.etiantian.org (10.0.0.8) 56(84) bytes of data.

64 bytes from www.etiantian.org (10.0.0.8): icmp\_seq=1 ttl=64 time=0.190 ms

64 bytes from www.etiantian.org (10.0.0.8): icmp\_seq=2 ttl=64 time=0.061 ms

[root@web01 conf]# ping bbs.etiantian.org ###ping一下两个域名，发现返回的IP就是服务器的IP，表示正常。

PING www.etiantian.org (10.0.0.8) 56(84) bytes of data.

64 bytes from www.etiantian.org (10.0.0.8): icmp\_seq=1 ttl=64 time=0.049 ms

[root@web01 conf]# curl [www.etiantian.org](http://www.etiantian.org) ###到此为止，基于域名的虚拟主机就配置完成了。

www.etiantian.org

[root@web01 conf]# curl bbs.etiantian.org

bbs.etiantian.org



当配置多个虚拟主机的时候，在浏览器中输入ip的时候，会弹出第一个虚拟机。为什么要找第一个呢？

http是有请求、响应报文，如果在浏览器里输入了[www.etiantian.org](http://www.etiantian.org)，请求报文的请求头会有一行host字段表示客户端告诉服务器想要这个主机的内容，此内容到达服务器nginx就会读取nginx.conf的配置文件，查看配置文件发现有www，所以就查找www对应的内容，发送回来。如果没有带host，就直接在配置文件中找第一个。

**基于域名的虚拟主机配置步骤：**

www.etiantian.org=======>html/www，浏览域名显示www.etiantian.org

bbs.etiantian.org=======>html/bbs，浏览域名显示bbs.etiantian.org

**a．配置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/www;

index index.html index.htm;

}

}

server {

listen 80;

server\_name bbs.etiantian.org;

location / {

root html/bbs;

index index.html index.htm;

}

}

}

~

**b．站点目录还不存在**

mkdir ../html/{www,bbs} -p

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

echo "bbs.etiantian.org" > ../html/bbs/index.html

**c.检查语法，重新加载nginx**

/application/nginx/sbin/nginx –t

/application/nginx/sbin/nginx –s reload

**d．配置hosts，测试**

Linux client：

[root@web01 conf]# tail -1 /etc/hosts

10.0.0.8 www.etiantian.org bbs.etiantian.org

curl www.etiantian.org

curl bbs.etiantian.org

如果是Windows：

C:\Windows\System32\drivers\etc\hosts

10.0.0.8 www.etiantian.org bbs.etiantian.org

浏览器里访问www.etiantian.org bbs.etiantian.org 域名。

实战：

增加blog.etiantian.org=====>html/bolg，浏览器域名显示blog.etiantian.org

**基于端口的虚拟主机配置实战**

基于端口的虚拟主机在生产环境中不多见，仅偶尔会用到，一般是为公司内部人员提供访问的，如OA系统、网站程序的后台、CMS发布后台、MySQL的web客户端PHPmyadmin等，使用特殊端口多是从安全上考虑的。下面演示的基于端口的虚拟主机相关配置部署

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 8001; ##将配置文件中的端口号修改。

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

}

}

server {

listen 8002;

server\_name bbs.etinatian.org;

location / {

root html/bbs;

index index.html index.htm;

}

}

server {

listen 8003;

server\_name blog.etinatian.org;

location / {

root html/blog;

"nginx.conf" 34L, 759C written

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

[root@web01 conf]# netstat -lnutup|grep nginx

tcp 0 0 0.0.0.0:8001 0.0.0.0:\* LISTEN 8424/nginx

tcp 0 0 0.0.0.0:8002 0.0.0.0:\* LISTEN 8424/nginx

tcp 0 0 0.0.0.0:8003 0.0.0.0:\* LISTEN 8424/nginx

[root@web01 conf]# curl www.etiantian.org

curl: (7) couldn't connect to host

[root@web01 conf]# curl www.etiantian.org:8001

www.etiantian.org

[root@web01 conf]# curl www.etiantian.org:8002

bbs.etiantian.org

[root@web01 conf]# curl www.etiantian.org:8003

blog.etiantian.org

**####包含，include####**

未来工作中，nginx配置文件会有上百上千行内容，为了方便运维人员查看更方便，故而优化，操作如下：

它可以放置nginx配置中任何位置，用法示例如下：

include mime.types;

include www.conf; ##<==包含单个文件

include extra/\*.conf; ##<==包含extra下所有以conf结尾的文件。

[root@web01 conf]# cat nginx.conf.haha ###将配置文件优化为如下

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

include extra/www.conf;

include extra/bbs.conf;

include extra/blog.conf;

}

[root@web01 conf]# mkdir extra ###因而配置文件中有次目录，故创建一个目录使用

[root@web01 conf]# sed -n '10,17p' nginx.conf.base-name #将www配置文件显示出来

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

}

}

[root@web01 conf]# sed -n '10,17p' nginx.conf.base-name > extra/www.conf #将www的配置文件放入到www.conf文件中。

[root@web01 conf]# sed -n '18,25p' nginx.conf.base-name >extra/bbs.conf #将bbs的配置文件放入到www.conf文件中。

[root@web01 conf]# sed -n '26,33p' nginx.conf.base-name >extra/blog.conf #将blog的配置文件放入到www.conf文件中。

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

[root@web01 conf]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

10.0.0.8 www.etiantian.org bbs.etiantian.org blog.etiantian.org

[root@web01 conf]# curl www.etiantian.org

www.etiantian.org

[root@web01 conf]# curl bbs.etiantian.org

www.etiantian.org

[root@web01 conf]# curl blog.etiantian.org

www.etiantian.org

**nginx status介绍**

nginx软件的功能模块中有一个ngx\_http\_stub\_status\_module模块，这个功能是记录nginx的基本访问状态信息，可以让我们了解nginx的工作状态。

[root@web01 conf]# cat extra/status.conf

##status

server {

listen 80;

server\_name status.etiantian.org;

location / {

stub\_status on;

access\_log off;

}

}

[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/www.conf;

include extra/bbs.conf;

include extra/blog.conf;

include extra/status.conf;

}

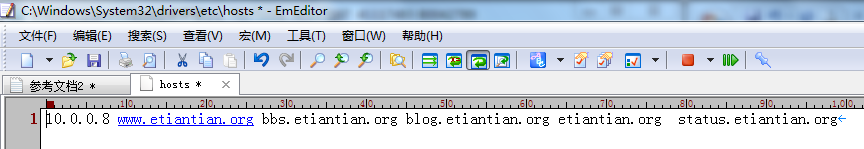
在window中的配置文件也要解析它，首先打开运行，输入drivers



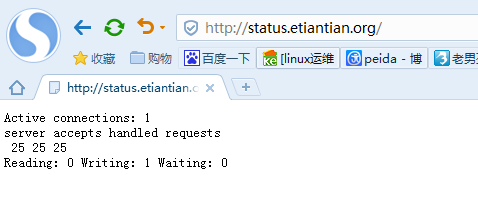
###进入etc文件，打开hosts文件



####将status.etiantian.org添加进去



###此时，在浏览器中输入status.etiantian.org时，反馈的结果如下



连接状态信息：

active connections:1

#<===表示nginx正处理的活动连接数1个。

server accepts handled requests

25 25 25

第一个server表示nginx启动到现在共处理了25个连接。

第二个accepts表示nginx启动到现在共成功了创建25此握手。

请求丢失数=（握手数-连接数），可以看出，本次状态显示没有丢失请求。

第三个handled requests 表示总共处理了25此请求。

reading：nginx 读取到客户端的header信息数。

writing：nginx 反馈给库互动的header信息数。

waiting：nginx 已经处理完正在等候下一次请求指令的驻留连接，开启keep-alive的情况下，这个值等于active-（reading+writing）。

**为nginx增加错误日志（error\_log）配置**

nginx软件会把自身运行的故障信息及用户访问的日志信息记录到指定的日志文件里。

1、nginx错误日志信息介绍

nginx的错误信息是调试nginx服务的重要手段，属于核心功能模块（ngx\_core\_module）的参数，该参数名字为error\_log，可以放在main区块中全局配置，也可以放置不同的虚拟主机中单独记录虚拟主机的错误信息。

error\_log的语法格式及参数语法说明如下：



其中，关键字error\_log不能改变，日志文件可以指定任意存放日志的目录，错误日志常见的有[debug|info|notice|warn|error|crit|alert|emerg]，级别越高记录的信息越少，场景一般是warn|error|crit这三个级别之一，注意不要配置info等较低级别，会带来磁盘I/O消耗。

error\_log的默认值为：

#default： error\_log logs/error.log error;

可以放置的标签段为：

# context: main http, server, location

nginx错误日志配置

编辑主配置文件nginx.conf，增加错误日志的配置，具体方法如下：

[root@web01 conf]# cat nginx.conf

worker\_processes 1;

error\_log logs/error.log error; #<====非常简单，一般配置这一行即可。

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

include extra/www.conf;

include extra/bbs.conf;

include extra/blog.conf;

include extra/status.conf;

}

[root@web01 conf]#

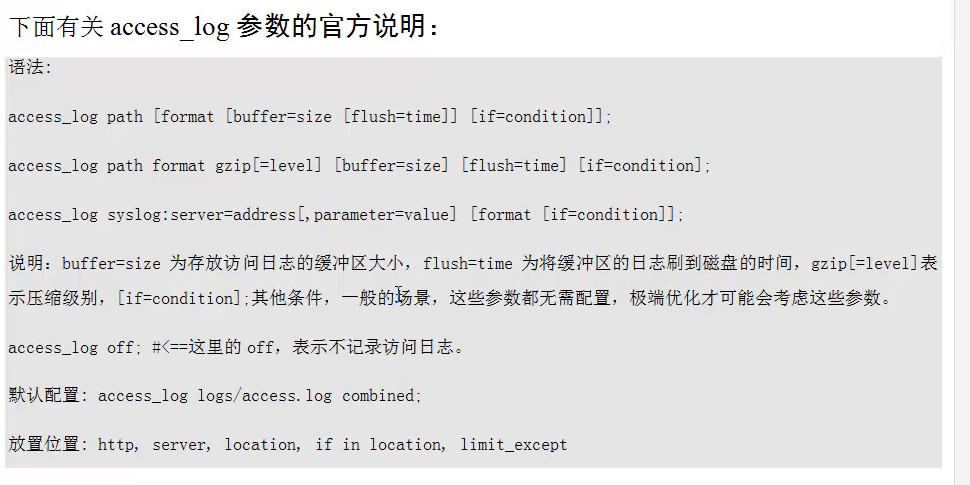
**nginx访问日志介绍**

nginx软件会把每个用户访问网站的日志信息记录到指定的日志文件里，供网站提供者分析用户浏览行为等，此功能由ngx\_http\_log\_module模块负责。

访问日志参数

nginx的访问日志主要由两个参数控制

|  |  |
| --- | --- |
| **参数** | **说明** |
| Log\_format | 用来定义记录日志的格式（可以定义多种日志格式，取不同名字即可） |
| Access\_log | 用来指定日志文件的路径及使用的何种日志格式记录日志。 |



[root@web01 conf]# vim nginx.conf.default

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

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

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

#####将此内容放入到nginx.conf文件中：

[root@web01 conf]# vim nginx.conf

worker\_processes 1;

error\_log logs/error.log error;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

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

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

#nginx vhosts config

include extra/www.conf;

include extra/bbs.conf;

include extra/blog.conf;

include extra/status.conf;

}

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

|  |  |
| --- | --- |
| **Nginx日志变量** | **说明** |
| $remote\_addr | 记录访问网站的客户端地址 |
| $http\_x\_forwarded\_for | 当前端有代理服务器时，设置web节点记录客户端地址的配置，此参数生效的前提是代理服务器上也要进行相关的x\_forwarded\_for设置。 |
| $remote\_user | 远程客户端用户名称。 |
| $time\_local | 记录访问时间与时区。 |
| $request | 用户的http请求起始行信息。 |
| $status | http状态码，记录请求返回的状态，例如：200、404、301等。 |
| $body\_bytes\_sents | 服务器发送给客户端的响应body字节数。 |
| $http\_referer | 记录此请求是从哪个链接访问过来的，可以根据erferer进行防盗链设置。 |
| $http\_user\_agent | 记录客户端访问信息，例如：浏览器，手机客户端等。 |

没有特殊要求，默认的配置即可，更多可以设置的记录日志信息的变量见：

http://nginx.org/en/docs/http/ngx\_http\_log\_module.html

开始配置www文件中的内容：

[root@web01 ~]# cd /application/nginx/conf

[root@web01 conf]# vi extra/www.conf

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

}

access\_log logs/access\_www.log main; ###将内容添加到这里。

}

"extra/www.conf" 9L, 230C written

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

[root@web01 conf]#

[root@web01 conf]# tail -f ../logs/access\_www.log ###查看日志的最新动态

此时打开浏览器输入IP：10.0.0.8时，文件内容会更新



10.0.0.1 - - [18/Nov/2015:12:55:54 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36 SE 2.X MetaSr 1.0" "-"

此时创建一个超链接：

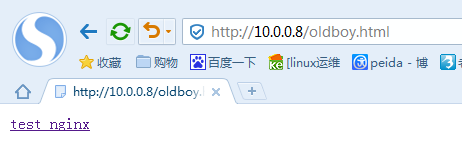
[root@web01 conf]# cd ../html/www/

[root@web01 www]# vi oldboy.html

<a href="http://10.0.0.8/">test nginx</a>

"oldboy.html" [New] 1L, 42C written

此时在浏览器中输入：10.0.0.8/oldboy.html后显示信息为：



点击一下test.nginx，会自动跳转到首页



[root@web01 www]# tail -f ../logs/access\_www.log

10.0.0.1 - - [18/Nov/2015:13:03:33 +0800] "GET /oldboy.html HTTP/1.1" 200 42 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36 SE 2.X MetaSr 1.0" "-"

**nginx日志轮询**

[root@web01 conf]# cd /server/scripts/

[root@web01 scripts]# vim cut\_nginx\_log.sh

#!/bin/sh

Dateformat=`date +%Y%m%d` ###定义的时间一天

Basedir="/application/nginx" ###nginx的安装目录

Nginxlogdir="$Basedir/logs" ###存放日志的目录

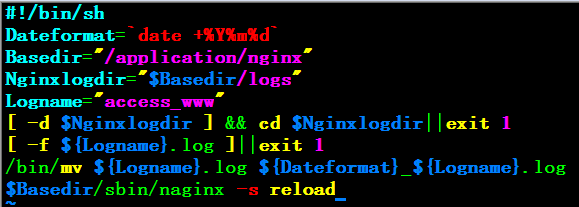
Logname="access\_www" ###日志的名字

[ -d $Nginxlogdir ] && cd $Nginxlogdir||exit 1 ###判断，如果存在目录，就切进去，否则就退出。

[ -f ${Logname}.log ]||exit 1 ###如果存在log不成立就退出

/bin/mv ${Logname}.log ${Dateformat}\_${Logname}.log ###把当前日志文件的名字改成想要的名字。

$Basedir/sbin/naginx -s reload ###改完名字之后原来的日志文件名就没有了，重启nginx后自动又生成一个名字。



[root@web01 scripts]# sh cut\_nginx\_log.sh

[root@web01 scripts]# ll /application/nginx/logs/

total 20

-rw-r--r-- 1 root root 3276 Nov 18 13:08 20151118\_access\_www.log

-rw-r--r-- 1 root root 4556 Nov 18 10:36 access.log

-rw-r--r-- 1 root root 0 Nov 18 13:27 access\_www.log

-rw-r--r-- 1 root root 653 Nov 18 13:27 error.log

-rw-r--r-- 1 root root 5 Nov 16 17:04 nginx.pid

[root@web01 scripts]# date -s "2015-11-16"

Mon Nov 16 00:00:00 CST 2015

[root@web01 scripts]# sh cut\_nginx\_log.sh

[root@web01 scripts]# ll /application/nginx/logs/

total 16

-rw-r--r-- 1 root root 0 Nov 18 2015 20151116\_access\_www.log

-rw-r--r-- 1 root root 0 Nov 18 2015 20151118\_access\_www.log

-rw-r--r-- 1 root root 4556 Nov 18 2015 access.log

-rw-r--r-- 1 root root 0 Nov 16 00:00 access\_www.log

-rw-r--r-- 1 root root 775 Nov 16 00:00 error.log

-rw-r--r-- 1 root root 5 Nov 16 2015 nginx.pid

[root@web01 scripts]# crontab –e ####将编辑好的脚本放入到定时任务中

#########nginx-by lidongkai 2015-11-18#####

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

此脚本可以优化一下：

#!/bin/sh

Dateformat=`date +%Y%m%d`

Basedir="/application/nginx"

Nginxlogdir="$Basedir/logs"

Logname1="access\_www"

Logname2="access\_bbs"

Logname3="access\_blog"

[ -d $Nginxlogdir ] && cd $Nginxlogdir||exit 1

[ -f ${Logname1}.log ]||exit 1

[ -f ${Logname2}.log ]||exit 1

[ -f ${Logname3}.log ]||exit 1

/bin/mv ${Logname1}.log ${Dateformat}\_${Logname1}.log

/bin/mv ${Logname2}.log ${Dateformat}\_${Logname2}.log

/bin/mv ${Logname3}.log ${Dateformat}\_${Logname3}.log

$Basedir/sbin/nginx -s reload

返回的结果为：

[root@web01 logs]# /bin/sh /server/scripts/nginx\_log.sh

[root@web01 logs]# ll

total 8

-rw-r--r-- 1 root root 0 Nov 18 14:57 20151118\_access\_bbs.log

-rw-r--r-- 1 root root 0 Nov 18 14:57 20151118\_access\_blog.log

-rw-r--r-- 1 root root 0 Nov 18 14:57 20151118\_access\_www.log

**nginx常用日志收集及分析工具**

nginx常用日志收集及分析工具有rsyslog、awstats、flume、ELK（Elasticsearch logstash kibanascribe）、storm等。更多常用开源运维工具：<http://oldboy.blog.51cto.com/2561410/775056>

nginx的rewrite功能

[root@web01 extra]# vim www.conf

server {

listen 80;

server\_name etiantian.org;

rewrite ^/(.\*) http://www.etiantian.org/$1 permanent;

}

####添加如上几行实现rewrite功能。

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

}

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

}

[root@web01 extra]# curl -I etiantian.org/oldboy.html

HTTP/1.1 301 Moved Permanently

Server: nginx/1.6.3

Date: Wed, 18 Nov 2015 07:50:04 GMT

Content-Type: text/html

Content-Length: 184

Connection: keep-alive

Location: http://www.etiantian.org/oldboy.html

**nginx rewrite的企业应用场景：**

nginx的rewrite功能在企业里应用非常广泛：

1、可以调整用户浏览的URL，看起来更规范，合乎开发及产品人员的需求。

2、为了让搜索引擎收录网站及用户体验更好，企业会将动态URL地址伪装成静态地址提供服务。

3、网站换新域名后，让旧的域名的访问跳转到新的域名上，例如：让京东的360buy换成了jd.com。

4、根据特殊变量、目录、客户端的信息进行URL跳转等。

**实现不同域名的URL跳转**

例1：实现访问<http://blog.etiantian.org>跳转到<http://www.etiantian.org/blog/oldboy.html>外部跳转时使用这种方法，浏览器地址会变为跳转后的地址，另外，要事先设置<http://www.eitiantian.org/blog/oldboy.html有结果输出，不然会出现401>等权限错误。

[root@web01 extra]# cat blog.conf

server {

listen 80;

server\_name blog.lifen.org;

location / {

root html/blog;

index index.html index.htm;

}

rewrite ^/blog/oldboy.html http://www.lifen.org/blog/oldboy.html break;

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

}

**nginx重要排错：**

将log中的pid误删除后，nginx是无法平滑重启的，如下所示：

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

nginx: [error] invalid PID number "" in "/application/nginx-1.6.3/logs/nginx.pid"

首先将所有nginx进程全部停止

[root@web01 application]# ps -ef|grep nginx

root 1081 834 0 14:40 pts/1 00:00:00 grep nginx

root 8424 1 0 00:33 ? 00:00:00 nginx: master process [root@web01 [root@web01 application]# ps -ef|grep nginx|grep -v gerp

root 1135 834 0 14:42 pts/1 00:00:00 grep nginx

[root@web01 application]# /application/nginx/sbin/nginx –c ##执行此命令后即可生效 /application/nginx/conf/nginx.conf

# 3、MySQL数据库

什么是数据库？存放数据的仓库。数据库有很多种类型，按照方向来分，传统的关系型数据库有：MySQL、oracle、db2、sysbase。互联网企业最火的就是MySQL与MariaDB，关系型数据库：本质上的二维表，服务器起一个数据库服务:www:t1,t2,t3… bbs:t1,t2,t3….blog:t1,t2,t3。

小结：什么是关系型数据库。

1、二维表格

2、MySQL和oracle数据库，运维最常用的是MySQL。

3、通过SQL结构化查询语句存取数据。

4、保持数据一致性方面很强。ACID理论。

特点：读写更多的是和磁盘打交道，数据一致性，安全

缺点：速度慢。

非关系型数据库：

nosql、not only sql，以高效，高性能为目的，凡是和效率性能无关的因素都尽可能抛弃。

是关系型数据库的补充，非革命。越来越火！memcached纯内存缓存的软件。

类型更多：

key—value 键值对形势stu001🡪oldboy

redis内存加持久化软件（磁盘）

数据类型：key🡪values键值对形式stu001🡪oldboy

类型更多：

key🡪values，集合set，列表list。。。。

NOSQL非关系型数据库小结：

1、NOSQL不是否定关系数据库，而是作为关系数据库的一个重要补充。

2、NOSQL为了高性能、高并发而生的。

3、NOSQL典型产品memcached（纯内存），redis（持久化缓存），mongodb（文档的数据库）

为什么选择MySQL数据库

毫无疑问，既成事实，绝大多数的使用Linux操作系统的大中小互联网网站都在使用MySQL作为其后端的数据库存储，从大兴的BAT门户，到电商平台，分类门户等无一例外使用MySQL数据库，那么，MySQL数据库到底有哪些优势和特点，让我们义无反顾选择它呢？

原因可能有一下几点：

1、MySQL性能卓越，服务稳定，很少出现异常宕机。

2、MySQL开放源代码且无版权制约，自主性及使用成本低。

3、MySQL历史悠久，社区及用户非常活跃，遇到问题，可以寻求帮助。

4、MySQL软件体积小，安装使用简单，并且易于维护，安装及维护成本低。

5、MySQL品牌口碑效应，使得企业无需考虑就直接用之，lamp、lemp流行架构。

6、MySQL支持多种操作系统，提供多种API接口，支持多种开发语言，特别对流行的PHP语言有很好的支持。

**MySQL家谱：**

第一条最正中产品线：根正苗红

版本：4.0—5.0—5.1—5.2

第二条产品线：

从5.4.xx开始-到了5.7.xx产品线系列介绍

为了更好地整合MySQL，AB公司社区和第三方公司开发的新存储引擎，以及吸收新的实现算法等，从而更好地支持SMP架构，提高性能而做了大量的代码重构。版本编号为5.4.xx开始，目前发展到了5.6.x。

\*MySQL 5.5 latest general availability （production）release

主流：互联网公司用MySQL 5.5，5.6

第三条产品线：

6.0.xx-到7.1.xx产品线系列介绍

为了更好地推广MySQL cluster版本，以及提高MySQL cluster的性能和稳定性，以及功能改进和增加，以及改动MySQL基础功能，使其对cluster存储引擎提供更有效的支持与优化。版本号为6.0.xx开发，目前发展到7.1.xx

安装MySQL：

Linux软件的安装方式：

1、yum/rpm 简单、块，无法定制。

2、编译安装，./configure;make;make install

缺点：复杂，速度慢；优点：可定制。

针对MySQL，第一条产品线的编译方式5.0—5.1

针对MySQL5.5以上版本使用编译安装，./cmake；gmake；gmake install

3、二进制包\*\*\*\*\*

解压就可使用（相当于绿色软件，无需安装）

特点：简单、快、不太好定制。

选择：二进制包

MySQL-5.5.32-Linux2.6-x86\_64.tar.gz

大网站做法：

1、先定制RPM包。

2、放到自习的yum仓库。

开始安装：

[root@web01 tools]# cd

[root@web01 ~]# cd /home/oldboy/tools/ #切换到oldboy家目录的tools目录下

[root@web01 tools]# rz –y #将压缩包放入到该目录下

rz waiting to receive.

zmodem trl+C ȡ

100% 182346 KB 5065 KB/s 00:00:36 0 Errorss86\_64.tar.gz...

[root@web01 tools]# ll

total 183140

-rw-r--r-- 1 root root 186722932 Jun 20 2013 mysql-5.5.32-linux2.6-x86\_64.tar.gz

drwxr-xr-x 9 1001 1001 4096 Nov 16 16:48 nginx-1.6.3

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

[root@web01 tools]# id mysql

id: mysql: No such user

[root@web01 tools]# useradd -s /sbin/nologin -M mysql #创建MySQL用户

[root@web01 tools]# id mysql

uid=508(mysql) gid=508(mysql) groups=508(mysql)

[root@web01 tools]# tar xf mysql-5.5.32-linux2.6-x86\_64.tar.gz #解压压缩包

[root@web01 tools]# ls

mysql-5.5.32-linux2.6-x86\_64 nginx-1.6.3

mysql-5.5.32-linux2.6-x86\_64.tar.gz nginx-1.6.3.tar.gz

[root@web01 tools]# mv mysql-5.5.32-linux2.6-x86\_64 /application/mysql-5.5.32

##将该文件移动到/application目录下

[root@web01 tools]# ln -s /application/mysql-5.5.32/ /application//mysql #创建软链接

[root@web01 tools]# ls -l /application/

total 12

lrwxrwxrwx 1 root root 26 Oct 16 18:30 apache -> /application/apache2.2.23/

drwxr-xr-x 2 root root 4096 Oct 16 18:29 apache2.2.23

lrwxrwxrwx 1 root root 26 Nov 23 11:36 mysql -> /application/mysql-5.5.32/

drwxr-xr-x 13 root root 4096 Nov 23 11:28 mysql-5.5.32

lrwxrwxrwx 1 root root 25 Nov 16 16:55 nginx -> /application/nginx-1.6.3/

drwxr-xr-x 11 root root 4096 Nov 16 17:04 nginx-1.6.3

[root@web01 tools]# ll /application/mysql/scripts/mysql\_install\_db #此目录本身就存在。

-rwxr-xr-x 1 7161 wheel 14816 Jun 19 2013 /application/mysql/scripts/mysql\_install\_db

[root@web01 tools]# /application/mysql/scripts/mysql\_install\_db --basedir=/application/mysql/ --datadir=/application/mysql/data/ --user=mysql

##安装MySQL服务

Installing MySQL system tables...

OK

Filling help tables...

OK

To start mysqld at boot time you have to copy

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

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

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

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

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

Alternatively you can run:

/application/mysql//bin/mysql\_secure\_installation

which will also give you the option of removing the test

databases and anonymous user created by default. This is

strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:

cd /application/mysql/ ; /application/mysql//bin/mysqld\_safe &

You can test the MySQL daemon with mysql-test-run.pl ##此时数据库已初始化完成

cd /application/mysql//mysql-test ; perl mysql-test-run.pl

Please report any problems with the /application/mysql//scripts/mysqlbug script!

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

cp: overwrite `/etc/my.cnf'? y

[root@web01 mysql]# ll /application/mysql/bin/mysqld\_safe #数据库的配置文件

-rwxr-xr-x 1 7161 wheel 24743 Jun 19 2013 /application/mysql/bin/mysqld\_safe

[root@web01 mysql]# sed -i 's#/usr/local/mysql#/application/mysql#g' /application/mysql/bin/mysqld\_safe ##更改一下配置文件的路径

[root@web01 mysql]# /application/mysql/bin/mysqld\_safe & #开启数据库服务（&在后台启用）

[1] 7387

[root@web01 mysql]# 151123 13:12:49 mysqld\_safe Logging to '/application/mysql/data/web01.err'.

151123 13:12:49 mysqld\_safe Starting mysqld daemon with databases from /application/mysql/data

[root@web01 mysql]# lsof -i :3306 #MySQL主端口为3306

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

mysqld 7605 mysql 10u IPv4 108517 0t0 TCP \*:mysql (LISTEN)

[root@web01 mysql]# cp /application/mysql/bin/\* /usr/local/sbin/ #将MySQL放入环境变量中

[root@web01 mysql]# which mysql #此时数据库已安装完成

/usr/local/sbin/mysql

[root@web01 mysql]# mysql

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

Your MySQL connection id is 1

Server version: 5.5.32 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, 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 mysql]# cp support-files/mysql.server /etc/init.d/mysqld ##配置传统方式启动MySQL

[root@web01 mysql]# sed -i 's#/usr/local/mysql#/application/mysql#g' /etc/init.d/mysqld

[root@web01 mysql]# chmod +x /etc/init.d/mysqld

[root@web01 mysql]# killall mysqld

[root@web01 mysql]# 151123 13:32:29 mysqld\_safe mysqld from pid file /application/mysql/data/web01.pid ended

[root@web01 mysql]# killall mysqld

mysqld: no process killed

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

[root@web01 mysql]# killall mysqld

mysqld: no process killed

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

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

Starting MySQL.. SUCCESS!

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

Shutting down MySQL. SUCCESS!

[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 8563 mysql 10u IPv4 110671 0t0 TCP \*:mysql (LISTEN)

[root@web01 mysql]# chkconfig --list mysqld

service mysqld supports chkconfig, but is not referenced in any runlevel (run 'chkconfig --add mysqld')

[root@web01 mysql]# chkconfig mysqld on

[root@web01 mysql]# chkconfig --list mysqld ##将MySQL加入开机自启动

mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@web01 mysql]# mysqladmin -uroot password "123456" #设置密码

[root@web01 mysql]# mysql #已设置好密码了，如果不加密码会不允许登录。

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

[root@web01 mysql]# mysql -uroot -p123456

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

Your MySQL connection id is 3

mysql> quit ##退出

Bye

[root@web01 mysql]# mysqladmin -uroot -p123456 password "oldboy" #修改密码

[root@web01 mysql]# mysql -uroot -p123456 #由于已修改密码，所以无法登录。

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

[root@web01 mysql]# mysql -uroot –poldboy

[root@web01 mysql]# mysql -uroot –p ##为了安全，登录数据库时不要将密码带上。

Enter password:

mysql> show databases; ###查看数据库中有几个库

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| test |

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

4 rows in set (0.03 sec)

mysql> drop database test; ###删除test库

Query OK, 0 rows affected (0.03 sec)

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

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

3 rows in set (0.00 sec)

mysql> select user,host from mysql.user; ###查找；用户；”,”分隔符；主机；在哪儿查；user表。

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

| user | host |

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

| root | 127.0.0.1 |

| root | ::1 |

| | localhost |

| root | localhost |

| | web01 |

| root | web01 |

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

6 rows in set (0.00 sec)

mysql> drop user ''@'localhost';

Query OK, 0 rows affected (0.02 sec)

mysql> drop user 'root'@'::1'; ###删除其他无用的用户和主机

Query OK, 0 rows affected (0.01 sec)

mysql> select user,host from mysql.user; ###在MySQL使用命令时，结尾必须加;分号

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

| user | host |

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

| root | 127.0.0.1 |

| root | web01 |

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

2 rows in set (0.00 sec)

mysql>grant all on \*.\* to ‘root’@localhost identified by ‘123456’with grant option;flush privileges; ###创建用户命令

mysql> flush privileges; ###刷新使其生效。

Query OK, 0 rows affected (0.00 sec)

**二进制包安装MySQL数据库**

1、添加用户

useradd mysql –s /sbin/nologin –M

2、下载解压软件

tar xf mysql-5.5.32-linux2.6-x86\_64.tar.gz

mv mysql-5.5.32-linux2.6-x86\_64 /application/mysql-5.5.32

ln –s /application/mysql-5.5.32/ /application/mysql

操作到此步相当于编译安装make install之后

3、初始化数据库

/application/mysql/scripts/mysql\_install\_db

--basedir=/application/mysql/ --datadir=/application/mysql/data/--user=mysql

4、授权MySQL管理数据库文件

chown –R mysql.mysql /application/mysql/

5、生成MySQL配置文件

cp /application/mysql/support-files/my-small.cnf /etc/my.cnf

6、配置启动MySQL

sed –i ‘s#/usr/local/mysql#/application/mysql#g’

/application/mysql/bin/mysqlld\_safe

/application/mysql/bin/mysqld\_safe &

7、配置环境变量

vi /etc/profile

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

soure /etc/profile

8、登录测试

MySQL

9、设置及更改密码

MySQLadmin –uroot password “123456”

mysqladmin –uroot –p123456 password “oldboy”

10、配置传统方式启动MySQL

[root@web01 mysql]# cp support-files/mysql.server /etc/init.d//mysqld

[root@web01 mysql]# sed -i 's#/usr/local/mysql#/application/mysql#g' /etc/init.d/mysqld

[root@web01 mysql]# chmod +x /etc/init.d/mysqld

[root@web01 mysql]# pkill mysqld

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

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

Starting MySQL.. SUCCESS!

[root@web01 mysql]# chkconfig mysqld on

[root@web01 mysql]# chkconfig --list mysqld

mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

11、安全优化

删除test库

drop database test;

删除无用用户（保留root localhost）

drop user ‘用户’@‘主机’；

如果主机名有大写，或者特殊符号的话，使用drop是无法删除的,应该使用：

delete from mysql.user where user=”root” and host=”A”; <==>drop user ‘root’@’A’;

12、MySQL简单命令总结：

man,help ==========

help show databases;

查所有的库： show databases; <===>ls

切换库：use mysql; <===>cd /root

查表： show tables;

查看用户列表：select user,host form mysql.user;

查看当前用户：select user();<===>whoami

查看当前所在库： select database();<===>pwd

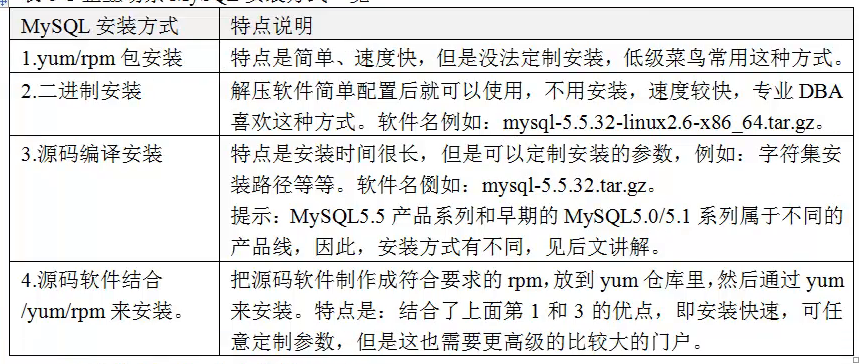
删除数据库：drop database 库名；

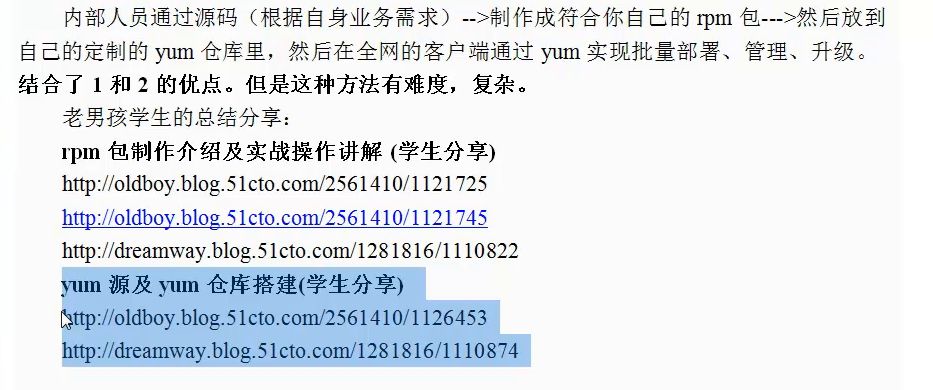
删除用户： drop user ‘用户’@’主机’;

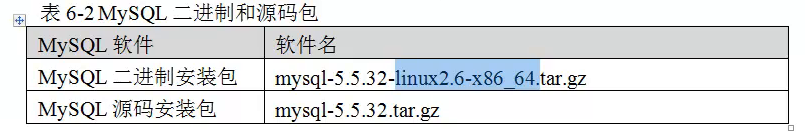
13、查看MySQL报错日志

[root@web01 data]# cat /application/mysql/data/web01.err

MySQL的安装方法也有很多，常见的方法有：







二进制包分为32位和64位。

# 4、FastCGI介绍

什么是CGI

CGI全称“通用网关接口”（common gateway interface），用于http服务器与其他机器上的程序服务通信交流的一种工具，CGI程序须运行在网络服务器上。

传统CGI接口方式的主要缺点是性能较差，因为每次http服务器遇到动态程序时都需要重新启动解析器来执行解析，然后结果被返回给HTTP服务器，这在处理高并发访问时，几乎是不可用的，因此就诞生了FastCGI，另外传统的CGI接口方式安全性也很差，现在已经很少被使用了。

什么是FastCGI

fastCGI是一个可伸缩的、高速的在HTTP服务器和动态脚本语言间通信的接口（FastCGI接口在Linux下是socket，（这个socket可以是文件socket，也可以是IP socket）），主要优点是把动态语言和HTTP服务器分离开来。多数流行的HTTP服务器都支持FastCGI，包括Apache、nginx和Lighttpd等。

同时，FastCGI也被许多脚本语言所支持，比较流行的脚本语言之一为PHP。FastCGI接口方式采用C/S架构，可以将HTTP服务器和脚本解析服务器分开，同时在脚本解析服务器上启动一个或多个脚本解析守护进程。当HTTP服务器每次遇到动态程序时，可以将其直接交付给FastCGI进程来执行，然后将得到的结果返回给浏览器。这种方式可以让HTTP服务器专一的处理静态或者将动态脚本服务器的结果返回给客户端，这在很大程度上提高了整个应用系统的性能。

FastCGI的重要特点总结：

1、FastCGI是HTTP服务器和动态脚本语言间通信的接口或者工具！

2、FastCGI优点是把动态语言解析和HTTP服务器分离开来。

3、nginx、Apache、Lighttpd以及多数动态语言都支持FastCGI。

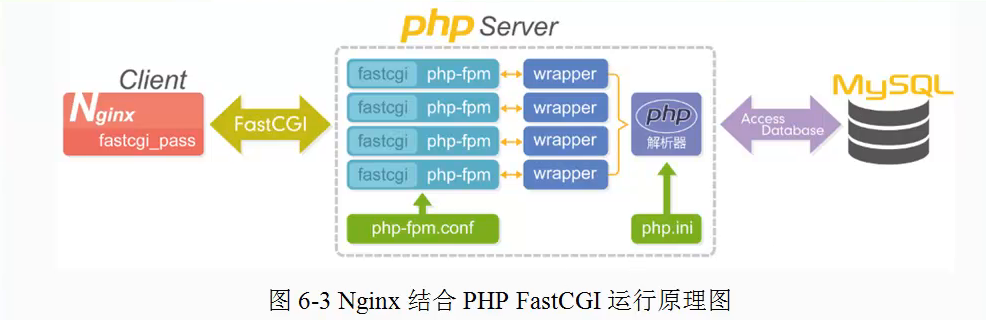
4、FastCGI接口方式采用C/S结构，分为客户端（HTTP服务器）和服务端（动态语言解析服务器）。

5、PHP动态语言服务端可以启动多个FastCGI的守护进程（例如：PHP-fpm（fcgi process mangement））。

6、http服务器通过（例如：nginx fastcgi\_pass）FastCGI客户端和动态语言FastCGI服务端通信（例如：PHP-fpm）。

nginx FastCGI的运行原理

nginx不支持对外部动态程序的直接调用或者解析，所有的外部程序（包括PHP）必须通过FastCGI接口来调用。FastCGI接口在Linux下是socket，（这个socket可以是文件socket也可以是IP socket）。为了调用CGI程序，还需要一个FastCGI的wrapper（wrapper可以理解为用于启动另一个程序的程序），这个wrapper绑定在某个固定socket上，如端口或者文件socket。当nginx将CGI请求发送给这个socket的时候，通过FastCGI接口，wrapper接受到请求，然后派生出一个新的线程，这个线程调用解析器或者外部程序处理脚本并读取返回数据：接着，wrapper再将返回的数据通过FastCGI接口，沿着固定的socket传递给nginx；最后，nginx将返回的数据发送给客户端，这就是nginx+ FastCGI的整个运作过程。详细的过程，如下图所示：



FastCGI的主要优点是把动态语言和HTTP服务器分离开来，使nginx专一处理静态请求和向后转发动态请求，而PHP/PHP-FPM服务器专一解析PHP动态请求。

# 5、安装PHP

[root@web01 data]# cd /home/oldboy/tools/

[root@web01 tools]# rz -y

rz waiting to receive.

zmodem trl+C ȡ

100% 14656 KB 14656 KB/s 00:00:01 0 Errors

yum install openssl-devel –y

[root@web01 tools]# rpm -qa zlib-devel libxml2-devel libjpeg-devel libiconv-devel

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

[root@web01 tools]# yum install -y zlib-devel libxm12-devel libjpeg-devel libiconv-devel

[root@web01 tools]# yum install -y freetype-devel libpng-devel gd-devel curl-devel libxslt-devel

[root@web01 tools]# rpm -qa zlib-devel libxml2-devel libjpeg-devel libiconv-devel

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

libxml2-devel-2.7.6-20.el6.x86\_64

[root@web01 tools]# rpm -qa freetype-devel libpng-devel gd-devel curl-devel

freetype-devel-2.3.11-15.el6\_6.1.x86\_64

libpng-devel-1.2.49-1.el6\_2.x86\_64

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

[root@web01 tools]# rpm -qa curl-devel

libcurl-7.19.7-46.el6.x86\_64

[root@web01 tools]# rpm -qa|grep libjpeg

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

[root@web02 tools]# rpm -qa zlib-devel libxml2-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

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

libpng-devel-1.2.49-1.el6\_2.x86\_64

libcurl-devel-7.19.7-46.el6.x86\_64

libxml2-devel-2.7.6-20.el6.x86\_64

freetype-devel-2.3.11-15.el6\_6.1.x86\_64

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

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

curl-devel <==>libcurl-devel-7.19.7-40.el6\_64.x86\_64

由于yum中没有libiconv-devel文件，所以要自己安装一下：

mkdir –p /home/oldboy/tools

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

make

make install

cd ../

安装libmcrypt库

wget <http://sourceforge.net/projects/mcrypt/files/Libmcrypt/2.5.8/libmcrypt-2.5.8.tar.gz>

tar zxf libmcrypt-2.5.8.tar.gz

cd libmcrypt-2.5.8

./configure

make

make install

sleep 2

/sbin/ldconfig

cd libltdl/

./configure –enable-ltdl-install

make

make install

cd ../../

可以选择 yum install –y libmcrypt-devel （但是官网没有这个文件，所以需要以下操作）

1、安装epel源

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

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

yum install -y libmcrypt-devel

rpm -qa libmcrypt-devel

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

安装mhash加密扩展库（同样需要epel源）

[root@web01 tools]# yum list|grep mhash

mhash.i686 0.9.9.9-3.el6 epel

mhash.x86\_64 0.9.9.9-3.el6 epel

mhash-devel.i686 0.9.9.9-3.el6 epel

mhash-devel.x86\_64 0.9.9.9-3.el6 epel

[root@web01 tools]# yum install -y mhash mhash-devel

[root@web01 tools]# rpm -qa mhash mhash-devel

mhash-devel-0.9.9.9-3.el6.x86\_64

mhash-0.9.9.9-3.el6.x86\_64

安装mcrypt加密扩展库

yum install -y mcrypt mcrypt-devel

rpm -qa mcrypt mcrypt-devel

mcrypt-2.6.8-10.el6.x86\_64

开始安装PHP（FastCGI方式）服务

[root@web01 tools]# tar xf php-5.3.27.tar.gz

[root@web01 tools]# cd php-5.3.27

./configure \

--prefix=/application/php5.3.27 \

--with-mysql=/application/mysql \ ###如果是服务器上没有nginx和MySQL的话，这里等号后面只写mysqlnd，在加一行--with-apxs2=/application/apache/bin/apxs \

--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-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-mhash \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-zip \

--enable-soap \

--enable-short-tags \

--enable-zend-multibyte \

--enable-static \

--with-xsl \

--with-fpm-user=nginx \ （这个位置选择apache）

--with-fpm-group=nginx \ （这个位置选择apache）

--enable-ftp

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

| License: |

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

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

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

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

| the installation process at this point. |

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

Thank you for using PHP.

[root@web01 php-5.3.27]# echo $?

0

make

make: \*\*\* [ext/phar/phar.php] Error 127 ##出现报错，使用如下命令解决：

[root@web01 php-5.3.27]# ln -s /application/mysql/lib/libmysqlclient.so.18 /usr/lib64/

[root@web01 php-5.3.27]# touch ext/phar/phar.hpar

[root@web01 php-5.3.27]# make

Build complete.

Don't forget to run 'make test'.

[root@web01 php-5.3.27]# make install ##安装结束后最后两行内容为：

/home/oldboy/tools/php-5.3.27/build/shtool install -c ext/phar/phar.phar /application/php5.3.27/bin

ln -s -f /application/php5.3.27/bin/phar.phar /application/php5.3.27/bin/phar

Installing PDO headers: /application/php5.3.27/include/php/ext/pdo/

[root@web01 php-5.3.27]# ll /application/php

lrwxrwxrwx 1 root root 23 Nov 23 20:07 /application/php -> /application/php5.3.27/

配置PHP服务解析配置.ini

[root@web01 php-5.3.27]# ll php.ini-\* ##第一个为开发环境，第二为生产环境，工作中选择后者，区别就是生产环境把错误关掉，开发环境把错误打开。

-rw-r--r-- 1 101 101 69606 Jul 11 2013 php.ini-development

-rw-r--r-- 1 101 101 69627 Jul 11 2013 php.ini-production

[root@web01 php-5.3.27]# cp php.ini-production /application/php/lib/php.ini ##将来要想优化php解析，就改这个配置文件。

配置PHP服务（fastCGI模式）配置文件PHP-fpm.conf

[root@web01 php-5.3.27]# cd /application/php/etc/

[root@web01 etc]# ls

pear.conf php-fpm.conf.default

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

[root@web01 etc]# ls

pear.conf php-fpm.conf php-fpm.conf.default

[root@web01 etc]# rz –y ###将修改好的配置文件上传到本目录中。

rz waiting to receive.

zmodem trl+C ȡ

100% 21 KB 21 KB/s 00:00:01 0 Errors

[root@web01 etc]# ls

pear.conf php-fpm.conf php-fpm.conf.default

[root@web01 etc]# /application/php/sbin/php-fpm ###启动PHP服务

[root@web01 etc]# ps -ef|grep php-fpm

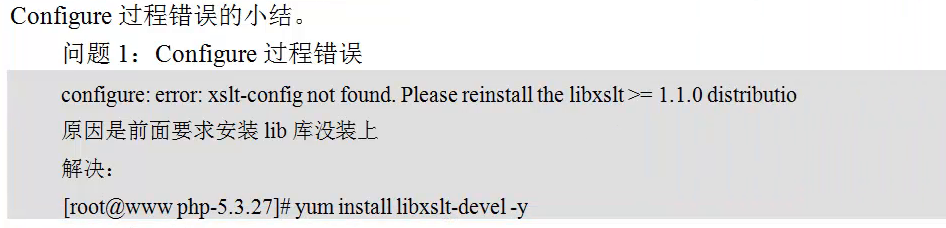
[root@web01 etc]# ps -ef|grep php-fpm|wc –l ###一个主进程，一个grep，16个子进程

18

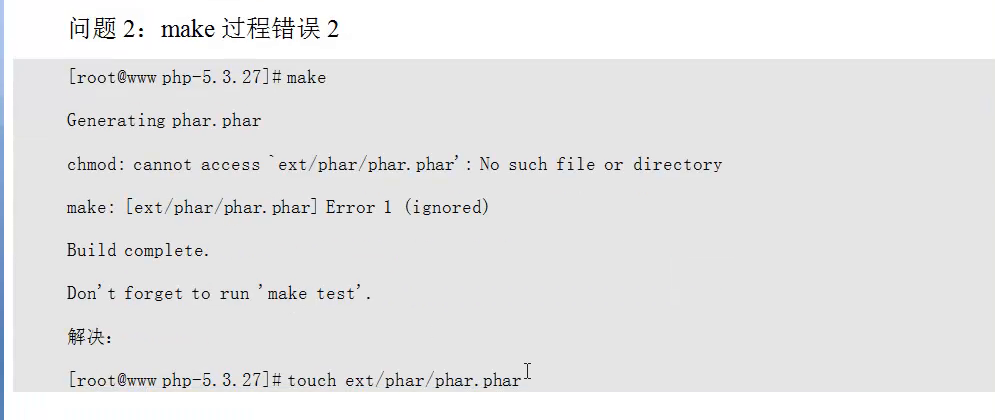
[root@web01 ~]# ss -lntup|grep php-fpm

tcp LISTEN 0 128 127.0.0.1:9000 \*:\* users:(("php-fpm",10371,7),("php-fpm",10372,0),("php-fpm",10373,0),("php-fpm",10374,0),("php-fpm",10375,0),("php-fpm",10376,0),("php-fpm",10377,0),("php-fpm",10378,0),("php-fpm",10379,0),("php-fpm",10380,0),("php-fpm",10381,0),("php-fpm",10382,0),("php-fpm",10383,0),("php-fpm",10384,0),("php-fpm",10385,0),("php-fpm",10386,0),("php-fpm",10387,0))

vi /etc/yum.conf







开始搭博客：

[root@web01 ~]# cd /application/nginx/conf/extra/

oot@web01 extra]# ll

total 16

-rw-r--r-- 1 root root 226 Nov 18 16:12 bbs.conf

-rw-r--r-- 1 root root 309 Nov 18 16:23 blog.conf

-rw-r--r-- 1 root root 144 Nov 18 16:12 status.conf

-rw-r--r-- 1 root root 359 Nov 18 16:17 www.conf

[root@web01 extra]# pwd

/application/nginx/conf/extra

[root@web01 extra]# vim blog.conf

1 server {

2 listen 80;

3 server\_name blog.lifen.org;

4 root html/blog;

5 location / {

6

7 index index.html index.htm;

8 }

9 location ~.\*\.(php|php5)?$ { ###添加此内容

10 fastcgi\_pass 127.0.0.1:9000;

11 fastcgi\_index index.php;

12 include fastcgi.conf;

13 }

14 access\_log logs/access\_blog.log main;

15 }

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

[root@web01 extra]# vim phpinfo.php

<?php

phpinfo();

?>

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

[root@web01 ~]# mysqladmin -uroot -p123456 password "oldboy123"

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

[root@web01 blog]# vim 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();

}

?>

"test\_mysql.php" [New] 8L, 228C written





# 6、安装wordpress博客

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

[root@web01 blog]# wget <https://cn.wordpress.org/wordpress-4.2.2-zh_CN.tar.gz>

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

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

3 rows in set (0.00 sec)

mysql> create database wordpress; ###创建库

Query OK, 1 row affected (0.00 sec)

mysql> show databases;

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| wordpress |

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

4 rows in set (0.00 sec)

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

Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql> select user,host from mysql.user where user='wordpress';

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

| user | host |

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

| wordpress | localhosts |

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

1 row in set (0.00 sec)

[root@web01 blog]# tar xf wordpress-4.2.2-zh\_CN.tar.gz

[root@web01 blog]# sz -y wordpress-4.2.2-zh\_CN.tar.gz

rz

zmodem trl+C ȡ

100% 6564 KB 6564 KB/s 00:00:01 0 Errorstar.gz...

[root@web01 blog]# ll

total 6584

-rw-r--r-- 1 root root 15 Nov 18 16:19 index.html

-rw-r--r-- 1 root root 20 Nov 23 21:39 phpinfo.php

-rw-r--r-- 1 root root 228 Nov 23 23:20 test\_mysql.php

drwxr-xr-x 5 nobody nfsnobody 4096 May 20 2015 wordpress

-rw-r--r-- 1 root root 6722406 May 20 2015 wordpress-4.2.2-zh\_CN.tar.gz

[root@web01 blog]# rm -f test\_mysql.php wordpress-4.2.2-zh\_CN.tar.gz

[root@web01 blog]# ll

total 12

-rw-r--r-- 1 root root 15 Nov 18 16:19 index.html

-rw-r--r-- 1 root root 20 Nov 23 21:39 phpinfo.php

drwxr-xr-x 5 nobody nfsnobody 4096 May 20 2015 wordpress

[root@web01 blog]# mv wordpress/\* .

[root@web01 blog]# ls

index.html wordpress wp-config-sample.php wp-load.php wp-trackback.php

index.php wp-activate.php wp-content wp-login.php xmlrpc.php

license.txt wp-admin wp-cron.php wp-mail.php

phpinfo.php wp-blog-header.php wp-includes wp-settings.php

readme.html wp-comments-post.php wp-links-opml.php wp-signup.php

[root@web01 blog]# rm -fr wordpress/

[root@web01 blog]# cd ../

[root@web01 html]# ll -ld blog/

drwxr-xr-x 5 root root 4096 Nov 23 23:52 blog/

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

[root@web01 html]# ll -ld blog/

drwxr-xr-x 5 nginx nginx 4096 Nov 23 23:52 blog/

[root@web01 html]# find ./blog/ -type f|xargs chmod 644

[root@web01 html]# find ./blog/ -type d|xargs chmod 755

[root@web01 html]# ll blog/

total 184

-rw-r--r-- 1 root root 15 Nov 18 16:19 index.html

-rw-r--r-- 1 root root 418 Sep 25 2013 index.php

-rw-r--r-- 1 root root 19930 Jan 1 2015 license.txt

-rw-r--r-- 1 root root 20 Nov 23 21:39 phpinfo.php

-rw-r--r-- 1 root root 6575 May 20 2015 readme.html

-rw-r--r-- 1 root root 4951 Aug 21 2014 wp-activate.php

drwxr-xr-x 9 root root 4096 May 20 2015 wp-admin

-rw-r--r-- 1 root root 271 Jan 9 2012 wp-blog-header.php

-rw-r--r-- 1 root root 5007 Jan 8 2015 wp-comments-post.php

-rw-r--r-- 1 root root 2898 May 20 2015 wp-config-sample.php

drwxr-xr-x 5 root root 4096 May 20 2015 wp-content

-rw-r--r-- 1 root root 3115 Mar 18 2015 wp-cron.php

drwxr-xr-x 12 root root 4096 May 20 2015 wp-includes

-rw-r--r-- 1 root root 2380 Oct 25 2013 wp-links-opml.php

-rw-r--r-- 1 root root 3123 Apr 13 2015 wp-load.php

-rw-r--r-- 1 root root 34241 Apr 13 2015 wp-login.php

-rw-r--r-- 1 root root 8252 Jul 17 2014 wp-mail.php

-rw-r--r-- 1 root root 11048 Apr 13 2015 wp-settings.php

-rw-r--r-- 1 root root 25152 Dec 1 2014 wp-signup.php

-rw-r--r-- 1 root root 4035 Dec 1 2014 wp-trackback.php

-rw-r--r-- 1 root root 3032 Feb 10 2014 xmlrpc.php

[root@web01 html]# mkdir blog/wp-content/uploads

[root@web01 html]# chown -R nginx.nginx blog/wp-content/uploads/

[root@web01 blog]# cd /application/nginx/conf/extra/

[root@web01 extra]# vi blog.conf

server {

listen 80;

server\_name blog.lifen.org;

root html/blog;

location / {

index index.php index.html index.htm;

}

"blog.conf" 14L, 396C written

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





为博客设置伪静态：

首先打开博客首页，点击设置--🡪固定连接---自定义结构



1.在WordPress后台-设置-固定链接-自定义结构，输入下面的代码，最后保存更改即可。

/archives/%post\_id%.html

https://codex.wordpress.org/Using\_Permalinks

http://codex.wordpress.org/zh-cn:%E4%BD%BF%E7%94%A8%E5%9B%BA%E5%AE%9A%E9%93%BE%E6%8E%A5

2、nginx配置文件的server容器中添加下面的代码

location / {

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

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

}

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

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

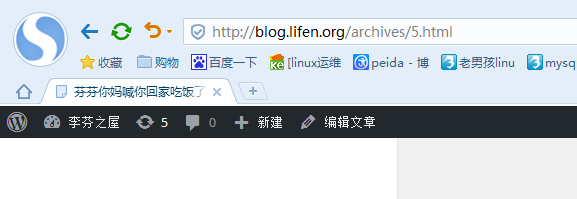
}

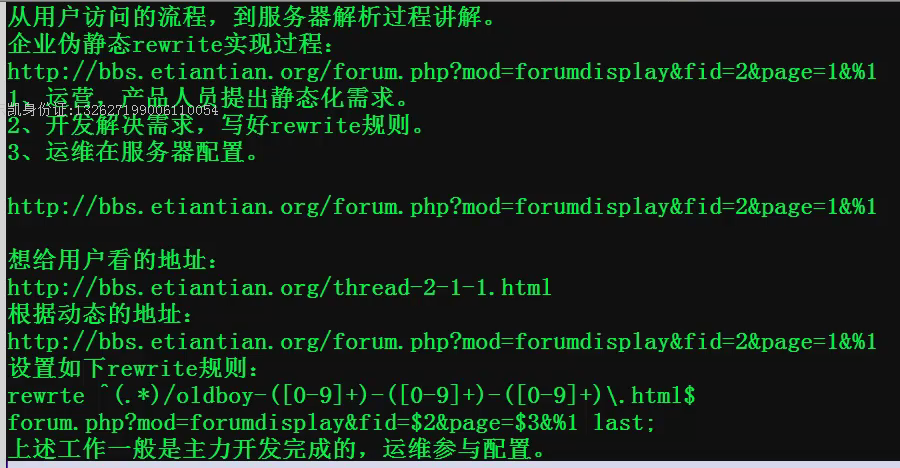
if (!-f $request\_filename){

rewrite (.\*) /index.php;

}

}





此时伪静态已经修改完成。

**数据库迁移**

迁移LNMP的数据库到独立的数据库服务器步骤：

（一）老服务器操作

1、从老的数据库里导出数据：

mysqldump -uroot -poldboy -A -B --events |gzip >/opt/bak.sql.gz

###-A 备份所有库 -B备份多个库，并添加use 库名：create database库等的功能。

2、拷贝到独立的数据库服务器

scp -P52113 /opt/bak.sql.gz oldboy@172.16.1.51:~

（二）新服务器操作

gzip -d bak.sql.gz

[root@db01 oldboy]# mysql -uroot -poldboy123</home/oldboy/bak.sql

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

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

| Tables\_in\_wordpress |

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

| ol\_commentmeta |

| ol\_comments |

| ol\_links |

| ol\_options |

| ol\_postmeta |

| ol\_posts |

| ol\_term\_relationships |

| ol\_term\_taxonomy |

| ol\_terms |

| ol\_usermeta |

| ol\_users |

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

（三）

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

Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> select user,host from mysql.user;

（四）web服务器上修改程序连接文件

修改：wp-config.php

连接主机改为db01.etiantian.org

做解析：

cat >/etc/hosts<<EOF

172.16.1.5 lb01

172.16.1.6 lb02

172.16.1.7 web02

172.16.1.8 web01

172.16.1.51 db01 db01.etiantian.org

172.16.1.31 nfs01

172.16.1.41 backup

172.16.1.61 m01

EOF

www静态程序服务软件Apache

Apache version

1.3 系列

2.0 系列

2.2 系列

2.4 系列

Apache软件包安装方式：

1）源码编译方式安装Apache：比较灵活，只编译想要的参数（中小公司常用）

##比喻自己做饭、做菜吃。

2）yum或RPM方式安装Apache：

优点：简单、方便，但是不够灵活。 ##比喻：下饭店吃饭。

a.功能和使用情况考虑，访问量小或内部使用的http的时候，多用yum install httpd -y

b.方便能能考虑的时候，可以yum安装。

c.维护成本考虑，服务器数量，比如几千台服务器安装Apache。

3）高级安装结合了编译和yum/rpm的双重优点：

通过源码（根据业务需求）-🡪制作成符合你自己的rmp-🡪放到自己的yum仓库里，然后在全网的客户端通过yum实现批量部署、管理、升级。

总结分享：

<http://oldboy.blog.51cto.com/2561410/1121725>

<http://oldboy.blog.51cto.com/2561410/1121745>

<http://dreamway.blog.51cto.com/1281816/1110822>

yum源及yum仓库搭建

<http://oldboy.blog.51cto.com/2561410/1126453>

<http://dreamway.blog.51cto.com/1281816/1110874>

Apache HTTP server（简称Apache）是Apache软件基金会的一个搞定能、功能强大，健壮可靠、又恨灵活的开放源代码的web服务软件，它可以运行在广泛使用的计算机平台上，如：Unix、Linux、Windows。因其多平台型和很好的安全性而被广泛使用，是当今互联网最流行的web服务端软件之一。

Apache的特点

功能强大、配置简单、速度快、应用广泛、性能稳定可靠，并可做代理服务器或负载均衡来使用。

Apache的应用场合

使用Apache运行静态HTML网页、图片（处理静态小文件能力不及nginx）

使用Apache结合PHP引擎运行PHP、Perl、Python等程序，lamp被称之为经典组合。

使用Apache结合tomcat/resin运行JSP、Java等程序，称为中小企业的首选。

使用Apache做代理、负载均衡、rewrite规则过滤等等。

安装Apache时，首先安装

[root@web02 ~]# rpm -qa httpd

httpd-2.2.15-47.el6.centos.x86\_64

安装过程：

cd /home/oldboy/tools

rz –y 加安装包

tar xf httpd-2.2.31.tar.gz

cd httpd-2.2.27

less install

less readme

yum install –y zlib zlib-devel

./configure --prefix=/application/apache2.2.31 --enable-deflate --enable-expires --enable-headers --enable-modules=most --enable-so --enable-rewrite --with-mpm=worker

make

make install

ln -s /application/apache2.2.31/ /application/apache

/application/apache/bin/apachectl –t

/application/apache/bin/apachectl start

[root@web02 tools]# lsof -i :80

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

httpd 6549 root 4u IPv6 60974 0t0 TCP \*:http (LISTEN)

httpd 6551 daemon 4u IPv6 60974 0t0 TCP \*:http (LISTEN)

httpd 6552 daemon 4u IPv6 60974 0t0 TCP \*:http (LISTEN)

httpd 6553 daemon 4u IPv6 60974 0t0 TCP \*:http (LISTEN)

[root@web02 tools]# ps -ef|grep httpd

root 6549 1 0 13:55 ? 00:00:00 /application/apache2.2.31/bin/httpd -k start

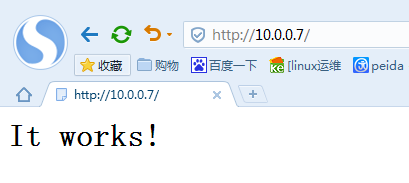
daemon 6550 6549 0 13:55 ? 00:00:00 /application/apache2.2.31/bin/httpd -k start

daemon 6551 6549 0 13:55 ? 00:00:00 /application/apache2.2.31/bin/httpd -k start

daemon 6552 6549 0 13:55 ? 00:00:00 /application/apache2.2.31/bin/httpd -k start

daemon 6553 6549 0 13:55 ? 00:00:00 /application/apache2.2.31/bin/httpd -k start

root 6637 1562 0 13:55 pts/0 00:00:00 grep httpd



[root@web02 tools]# /application/apache/bin/apachectl

Usage: /application/apache2.2.31/bin/httpd [-D name] [-d directory] [-f file]

[-C "directive"] [-c "directive"]

[-k start|restart|graceful|graceful-stop|stop]

[-v] [-V] [-h] [-l] [-L] [-t] [-T] [-S]

Options:

-D name : define a name for use in <IfDefine name> directives

-d directory : specify an alternate initial ServerRoot

-f file : specify an alternate ServerConfigFile

-C "directive" : process directive before reading config files

-c "directive" : process directive after reading config files

-e level : show startup errors of level (see LogLevel)

-E file : log startup errors to file

-v : show version number

-V : show compile settings

-h : list available command line options (this page)

-l : list compiled in modules

-L : list available configuration directives

-t -D DUMP\_VHOSTS : show parsed settings (currently only vhost settings)

-S : a synonym for -t -D DUMP\_VHOSTS

-t -D DUMP\_MODULES : show all loaded modules

-M : a synonym for -t -D DUMP\_MODULES

-t : run syntax check for config files

-T : start without DocumentRoot(s) check

[root@web02 apache]# cd htdocs/

[root@web02 htdocs]# ll

total 4

-rw-r--r-- 1 root root 44 Nov 21 2004 index.html ###为什么index.html为主站点文件？ 在apache的配置文件中查找DirectoryIndex就可得知。

[root@web02 htdocs]# cat index.html

<html><body><h1>It works!</h1></body></html>[root@web02 htdocs]#

[root@web02 htdocs]# rm -fr \*

[root@web02 htdocs]# vim index.html

<html>

<head><title>oldboy,s blog.</title></head>

<body>

Hi,I am oldboy. My blog address is

<a href="http://oldboy.blog.51cto.com"> targe=\_parent>http://oldboy.blog.51cto.com</

a>

</body>

</html>

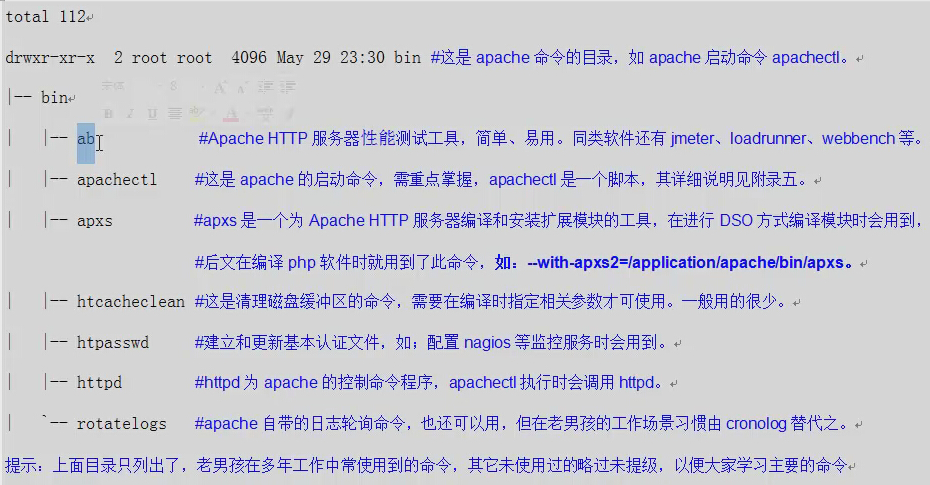
"index.html" 8L, 198C written

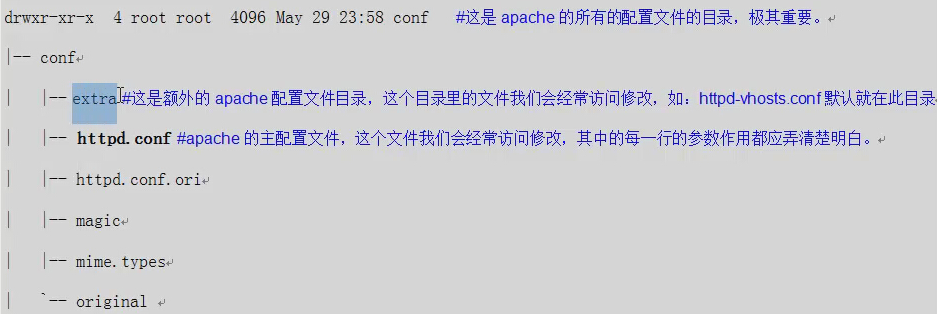


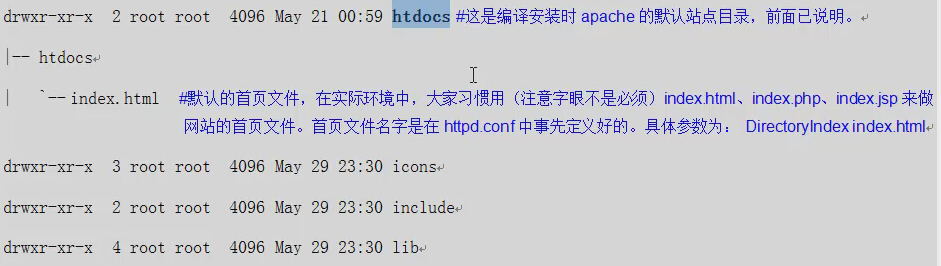
#######若不修改配置文件的名字则不用重新重启服务。

Apache平滑重启的命令：

/application/apache/bin/apachectl graceful



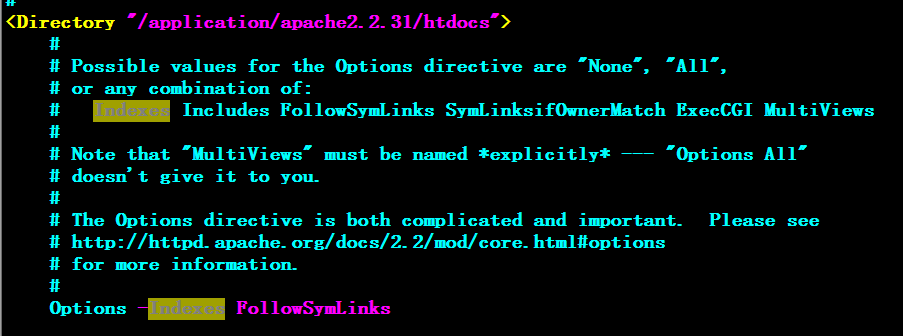






Apache优化

如果站点文件被更改，则在浏览器中输入IP地址时还是会显示出网站相应内容，而且会暴露信息，想不让信息暴露的话，要进行一个优化，在Apache配置文件，vim httpd.conf，将最后一行的第二个参数前面加一个“-”即可。





为优化后的结果

虚拟主机：

1、部署多个站点，每个站点，希望用不同的域名和站点目录，或者是不同的端口，不同的IP，需要虚拟主机功能。

一句话，一个http服务要配置多个站点，就需要虚拟主机。

虚拟机分类：

a.基于域名\*\*\*\*\*

b.基于端口

c.基于IP

开始测试：

测试内容为，三个不同的网站，使其建立基于域名、端口、IP的虚拟主机

域名与站点目录

[www.etiantia.org](http://www.etiantia.org) /var/html/www

blog.etiantian.org /var/html/blog

bbs.etiantian.org /var/html/bbs

[root@web02 apache]# mkdir -p /var/html/{www,blog,bbs}

[root@web02 apache]# tree /var/html/

/var/html/

├── bbs

├── blog

└── www

3 directories, 0 files

[root@web02 apache]# for name in www blog bbs;do echo "http://$name.etiantian.org" >/var/html/$name/index.html;done

[root@web02 apache]# for name in www blog bbs;do echo cat /var/html/$name/index.html;done cat /var/html/www/index.html

cat /var/html/blog/index.html

cat /var/html/bbs/index.html

[root@web02 apache]# tree /var/html/

/var/html/

├── bbs

│   └── index.html

├── blog

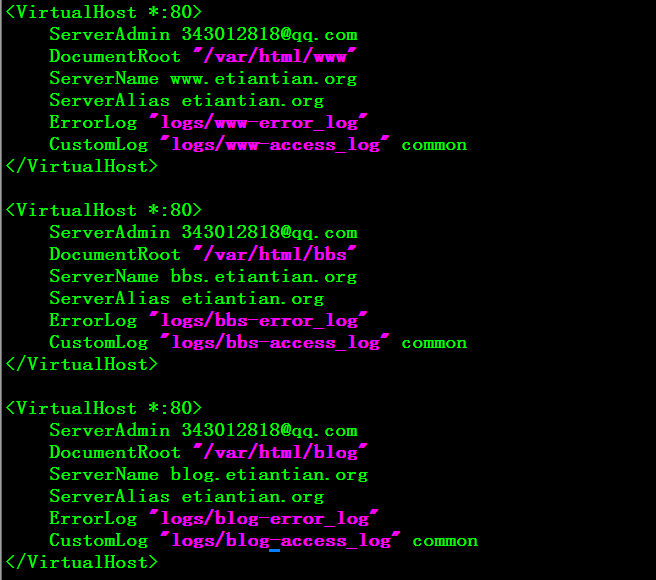
│   └── index.html

└── www

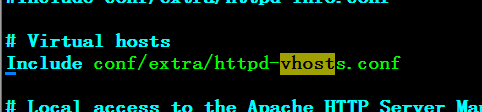
└── index.html

3 directories, 3 files

[root@web02 extra]# vim httpd-vhosts.conf



[root@web02 conf]# vim httpd.conf





将这两行的注释取消掉。

[root@web02 conf]# /application/apache/bin/apachectl -t

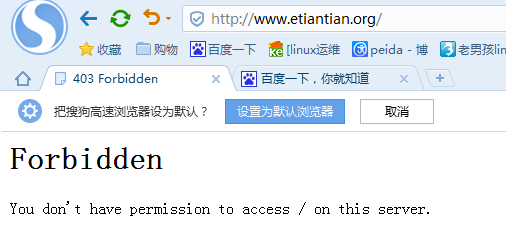
httpd: Could not reliably determine the server's fully qualified domain name, using 163.16.1.7 for ServerName

Syntax OK

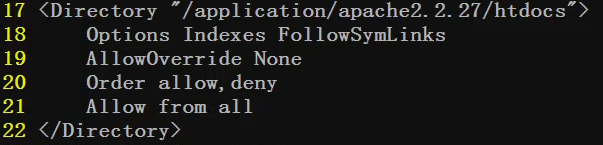
[root@web02 conf]# /application/apache/bin/apachectl graceful

httpd: Could not reliably determine the server's fully qualified domain name, using 163.16.1.7 for ServerName

此时配置文件已配置完成，服务也已重启完毕，可是浏览器返回的结果是



导致403错误的原因是主配置文件有一段内容，将这段内容放到主配置文件的最后一行然后进行修改。



<Directory "/var/html">

Options -Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

使用端口虚拟主机的配置

[root@web02 conf]# vim httpd.conf

# Listen: Allows you to bind Apache to specific IP addresses and/or

# ports, instead of the default. See also the <VirtualHost>

# directive.

#

# Change this to Listen on specific IP addresses as shown below to

# prevent Apache from glomming onto all bound IP addresses.

#

#Listen 12.34.56.78:80

Listen 80 ##在这个位置上添加两个端口，进行修改

Listen 81

Listen 82

[root@web02 extra]# vim httpd-vhosts.conf



<VirtualHost \*:80>

ServerAdmin oldboy@oldboyedu.com # 管理员邮箱

DocumentRoot "/application/apache2.2.31/html/bbs" # 网站的站点目录

ServerName bbs.etiantian.org # 域名

ErrorLog "/app/logs/bbs-error\_log" # 错误日志

CustomLog "/app/logs/bbs-access\_log" common # 访问日志

</VirtualHost>

将端口修改为上面的图片。

[root@web02 extra]# ss -lntup|grep 80

tcp LISTEN 0 511 :::80 :::\* users:(("httpd",6549,4),("httpd",14080,4),("httpd",14081,4))

tcp LISTEN 0 511 :::81 :::\* users:(("httpd",6549,6),("httpd",14080,6),("httpd",14081,6))

tcp LISTEN 0 511 :::82 :::\* users:(("httpd",6549,15),("httpd",14080,15),("httpd",14081,15))

基于IP的虚拟主机：

如果不想平滑重启时出现如下报错的话，请配置

[root@web02 ~]# /application/apache/bin/apachectl graceful

httpd: Could not reliably determine the server's fully qualified domain name, using 163.16.1.7 for ServerName

[root@web02 conf]# vim httpd.conf



**在apache服务器上安装PHP**

[root@web02 tools]# rpm -qa zlib-devel libxml2-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel

[root@web02 tools]#yum install –y zlib-devel libxml2-devel libjpeg-turbo-devel libiconv-devel freetype-devel libpng-devel gd-devel libcurl-devel

由于yum中没有libiconv-devel文件，所以要自己安装一下：

mkdir –p /home/oldboy/tools

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

make

make install

cd ../

安装libmcrypt库

可以选择 yum install –y libmcrypt-devel （但是官网没有这个文件，所以需要以下操作）

1、安装epel源

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

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

yum install -y libmcrypt-devel

rpm -qa libmcrypt-devel

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

安装mhash加密扩展库（同样需要epel源）

[root@web01 tools]# yum list|grep mhash

mhash.i686 0.9.9.9-3.el6 epel

mhash.x86\_64 0.9.9.9-3.el6 epel

mhash-devel.i686 0.9.9.9-3.el6 epel

mhash-devel.x86\_64 0.9.9.9-3.el6 epel

[root@web01 tools]# yum install -y mhash mhash-devel

[root@web01 tools]# rpm -qa mhash mhash-devel

mhash-devel-0.9.9.9-3.el6.x86\_64

mhash-0.9.9.9-3.el6.x86\_64

安装mcrypt加密扩展库

yum install -y mcrypt mcrypt-devel

rpm -qa mcrypt mcrypt-devel

mcrypt-2.6.8-10.el6.x86\_64

./configure \

--prefix=/application/php5.3.27 \

--with-apxs2=/application/apache/bin/apxs \

--with-xmlrpc \

--with-openssl \

--with-zlib \

--with-freetype-dir \

--with-gd \

--with-jpeg-dir \

--with-png-dir \

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

--enable-short-tags \

--enable-sockets \

--enable-zend-multibyte \

--enable-soap \

--enable-mbstring \

--enable-static \

--enable-gd-native-ttf \

--with-curl \

--with-xsl \

--enable-ftp \

--with-libxml-dir

--with-fpm-user=apache \

--with-fpm-group=apache \

make

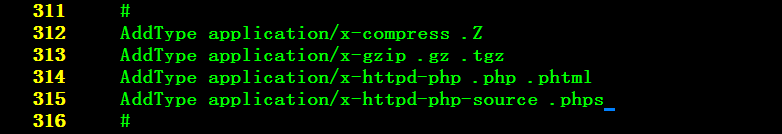
make install

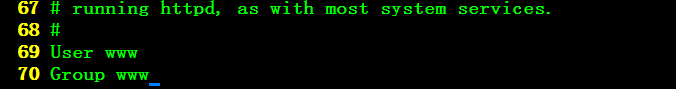
ln -s /application/php5.3.27/ /application/php

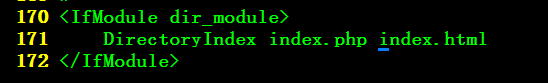
cp php.ini-production /application/php/lib/php.ini

**配置apache支持PHP只要3步如下所示：**

在[root@web02 conf]# vi httpd.conf配置文件中添加两行







[root@web02 ~]# id www

id: www: No such user

[root@web02 ~]# useradd -s /sbin/nologin -M www

[root@web02 ~]# id www

uid=507(www) gid=507(www) groups=507(www)

[root@web02 extra]# cd /var/html/blog/

[root@web02 blog]# ll

total 4

-rw-r--r-- 1 root root 26 Nov 25 18:09 index.html

[root@web02 blog]# vi index.php

<?php

phpinfo();

?>

"index.php" [New] 3L, 22C written

[root@web02 blog]# /application/apache/bin/apachectl -t

Syntax OK

[root@web02 blog]# /application/apache/bin/apachectl graceful

[root@web02 blog]# ll

total 8

-rw-r--r-- 1 root root 26 Nov 25 18:09 index.html

-rw-r--r-- 1 root root 22 Nov 26 13:00 index.php

[root@web02 blog]# cat index.php

<?php

phpinfo();

?>

[root@web02 blog]# rm -fr index.php

[root@web02 blog]# ll

total 4

-rw-r--r-- 1 root root 26 Nov 25 18:09 index.html

安装负载均衡



ip addr add 10.0.0.3/24 dev eth1

ip add|grep 0.4

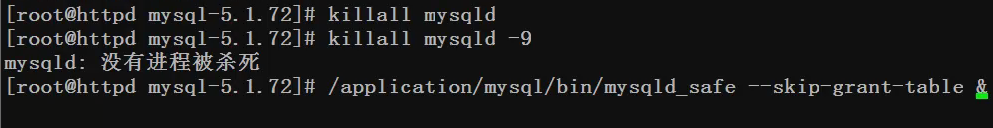
数据库报错信息：

[root@web01 blog]# mysqladmin -uroot -p123456 password oldboy123

mysqladmin: connect to server at 'localhost' failed

error: 'Access denied for user 'root'@'localhost' (using password: YES)'

[root@web01 ~]# killall mysqld



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

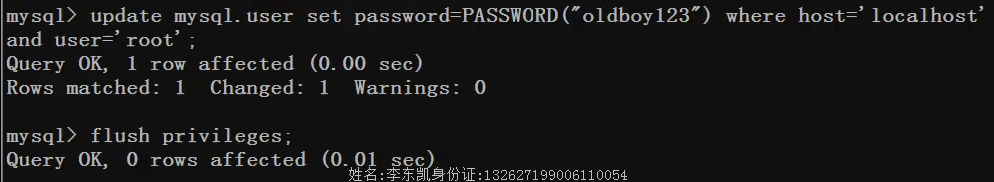
[root@web01 ~]# rm -fr /application/mysql/data/\*

[root@web01 ~]# /application/mysql/scripts/mysql\_install\_db --basedir=/application/mysql --datadir=/application/mysql/data/ --user=mysql

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

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

[root@web01 ~]# mysql



<?php

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

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

if($link\_id){

echo "mysql successful by oldboy !";

}else{

echo mysql\_error();

}

?>

安装PHP时发现的报错：

出错  
configure: error: Please reinstall the BZip2 distribution  
解决方法  
yum -y install bzip2 bzip2-devel  
make编译出错  
make: \*\*\* [sapi/cli/php] Error 1  
解决方法  
make clean  
make ZEND\_EXTRA\_LIBS='-liconv'  
make install