

第2天-Web 服务器

一、Web 服务器介绍

1、WEB服务简介

- 目前最主流的三个Web服务器是Apache、Nginx、IIS。
- Web服务器一般指网站服务器，是指驻留于因特网上某种类型计算机的程序，可以向浏览器等Web客户端提供文档，也可以放置网站文件，让全世界浏览；可以放置数据文件，让全世界下载。
- WEB服务器也称为WWW(WORLD WIDE WEB)服务器，主要功能是提供网上信息浏览服务。Web服务器是可以向发出请求的浏览器提供文档的程序：
- 服务器是一种被动程序只有当Internet上运行其他计算机中的浏览器发出的请求时，服务器才会响应



- 最常用的Web服务器是Apache，Nginx和Microsoft的Internet信息服务器（Internet Information Services，IIS）。
- Internet上的服务器也称为Web服务器，是一台在Internet上具有独立IP地址的计算机，可以向Internet上的客户机提供WWW、Email和FTP等各种Internet服务。
- Web服务器是指驻留于因特网上某种类型计算机的程序。当Web浏览器（客户端）连到服务器上并请求文件时，服务器将处理该请求并将文件反馈到该浏览器上，附带的信息会告诉浏览器如何查看该文件（即文件类型）。服务器使用HTTP（超文本传输协议）与客户机浏览器进行信息交流，这就是人们常把它们称为HTTP服务器的原因。
- Web服务器不仅能够存储信息，还能在用户通过Web浏览器提供的信息的基础上运行脚本和程序。

2、WEB 服务协议

- WEB 服务应用层使用HTTP协议。
- HTML（标准通用标记语言下的一个应用）文档格式。
- 浏览器统一资源定位器（URL）。
- 为了解决HTTP协议的这一缺陷，需要使用另一种协议：安全套接字层超文本传输协议HTTPS。为了数据传输的安全，HTTPS在HTTP的基础上加入了SSL协议，SSL依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信加密。WWW 采用的是浏览器/服务器结构，其作用是整理和储存各种WWW资源，并响应客户端软件的请求，把客户所需的资源传送到 Windows、UNIX 或 Linux 等平台上。

3、web 服务器与应用服务器的区别

- Web服务器传送 (serves) 页面使浏览器可以浏览, 然而应用程序服务器提供的是客户端应用程序可以调用 (call) 的方法 (methods)。确切一点, 你可以说: Web服务器专门处理HTTP请求 (request), 但是应用程序服务器是通过很多协议来为应用程序提供 (serves) 商业逻辑 (business logic)。
- Web服务器可以解析 (handles) HTTP协议。当Web服务器接收到一个HTTP请求 (request), 会返回一个HTTP响应 (response), 例如送回一个HTML页面。为了处理一个请求 (request), Web服务器可以响应 (response) 一个静态页面或图片, 进行页面跳转 (redirect), 或者把动态响应 (dynamic response) 的产生委托 (delegate) 给一些其它的程序例如CGI脚本, JSP (Java Server Pages) 脚本, servlets, ASP (Active Server Pages) 脚本, 服务器端 (server-side) JavaScript, 或者一些其它的服务器端 (server-side) 技术。无论它们 (译者注: 脚本) 的目的如何, 这些服务器端 (server-side) 的程序通常产生一个HTML的响应 (response) 来让浏览器可以浏览。
- Web服务器的代理模型 (delegation model) 非常简单。当一个请求 (request) 被送到Web服务器里来时, 它只单纯的把请求 (request) 传递给可以很好的处理请求 (request) 的程序。Web服务器仅仅提供一个可以执行服务器端 (server-side) 程序和返回 (程序所产生的) 响应 (response) 的环境, 而不会超出职能范围。服务器端 (server-side) 程序通常具有事务处理 (transaction processing), 数据库连接 (database connectivity) 和消息 (messaging) 等功能。
- 虽然Web服务器不支持事务处理或数据库连接池, 但它可以配置 (employ) 各种策略 (strategies) 来实现容错性 (fault tolerance) 和可扩展性 (scalability), 例如负载平衡 (load balancing), 缓冲 (caching)。集群特征 (clustering—features) 经常被误认为仅仅是应用程序服务器专有的特征。

4、WEB 服务器的工作原理

- Web服务器的工作原理并不复杂, 一般可分成如下4个步骤: 连接过程、请求过程、应答过程以及关闭连接。

1、连接过程: Web服务器和其浏览器之间所建立起来的一种连接。查看连接过程是否实现, 用户可以找到和打开socket这个虚拟文件, 这个文件的建立意味着连接过程这一步骤已经成功建立。

2、请求过程: Web的浏览器运用socket这个文件向其服务器而提出各种请求。

3、应答过程: 运用HTTP协议把在请求过程中所提出来的请求传输到Web的服务器, 进而实施任务处理, 然后运用HTTP协议把任务处理的结果传输到Web的浏览器, 同时在Web的浏览器上面展示上述所请求之界面。

4、关闭连接: 当上一个步骤--应答过程完成以后, Web服务器和其浏览器之间断开连接之过程。

Web服务器上述4个步骤环环相扣、紧密相联, 逻辑性比较强, 可以支持多个进程、多个线程以及多个进程与多个线程相混合的技术。

二、Apache 和 Nginx 的介绍

1、Apache 介绍

- Apache是世界使用排名第一的Web服务器软件。它可以运行在几乎所有广泛使用的计算机平台上, 由于其跨平台和安全性被广泛使用, 是最流行的Web服务器端软件之一。
- Apache源于NCSAhttpd服务器, 经过多次修改, 成为世界上最流行的Web服务器软件之一。Apache取自“a patchy server”的读音, 意思是充满补丁的服务器, 因为它是自由软件, 所以不断有人来为它开发新的功能、新的特性、修改原来的缺陷。Apache的特点是简单、速度快、性能稳定, 并可做代理服务器来使用。

- 到目前为止Apache仍然是世界上用的最多的Web服务器，市场占有率达60%左右。世界上很多著名的网站如Amazon、Yahoo!、W3 Consortium、Financial Times等都是Apache的产物，它的成功之处主要在于它的源代码开放、有一支开放的开发队伍、支持跨平台的应用（可以运行在几乎所有的Unix、Windows、Linux系统平台上）以及它的可移植性等方面。
- Apache 的特性:
 - 几乎可以运行在所有的计算机平台上.
 - 支持最新的http/1.1协议、
 - 简单而且强有力的基于文件的配置(httpd.conf).
 - 支持通用网关接口(cgi)
 - 支持虚拟主机.
 - 支持http认证.
 - 集成perl.
 - 集成的代理服务器
 - 可以通过web浏览器监视服务器的状态, 可以自定义日志.
 - 支持服务器端包含命令(ssi).
 - 支持安全socket层(ssl).
 - 具有用户会话过程的跟踪能力.
 - 支持fastcgi
 - 支持java servlets

2、nginx 介绍

- nginx是俄罗斯人编写的十分轻量级的HTTP服务器,它的发音为“engine X”，是一个高性能的HTTP和反向代理服务器，同时也是一个IMAP/POP3/SMTP 代理服务器．Nginx是由俄罗斯人 Igor Sysoev为俄罗斯访问量第二的 Rambler.ru站点开发.
- Nginx以事件驱动的方式编写，所以有非常好的性能，同时也是一个非常高效的反向代理、负载均衡。其拥有匹配 Lighttpd的性能，同时还没有Lighttpd的内存泄漏问题，而且Lighttpd的 mod_proxy也有一些问题并且很久没有更新。但是Nginx并不支持cgi方式运行，原因是可以减少因此带来的一些程序上的漏洞。所以必须使用FastCGI方式来执行PHP程序。
- nginx做为HTTP服务器，有以下几项基本特性：
 - 处理静态文件，索引文件以及自动索引；打开文件描述符缓冲．
 - 无缓存的反向代理加速，简单的负载均衡和容错．
 - FastCGI，简单的负载均衡和容错．
 - Nginx专为性能优化而开发，性能是其最重要的考量,实现上非常注重效率。它支持内核Poll模型，能经受高负载的考验,有报告表明能支持高达 50,000个并发连接数
 - Nginx具有很高的稳定性。其它HTTP服务器，当遇到访问的峰值，或者有人恶意发起慢速连接时，也很可能会导致服务器物理内存耗尽频繁交换，失去响应，只能重启服务器。例如当前apache一旦上到200个以上进程，web响应速度就明显非常缓慢了。而Nginx采取了分阶段资源分配技术，使得它的CPU与内存占用率非常低。nginx官方表示保持10,000个没有活动的连接，它只占2.5M内存，所以类似DOS这样的攻击对nginx来说基本上是毫无用处的。就稳定性而言,nginx比lighttpd更胜一筹。
 - Nginx支持热部署。它的启动特别容易,并且几乎可以做到7*24不间断运行，即使运行数个月也不需要重新启动。你还能够在不间断服务的情况下，对软件版本进行进行升级。

3、Apache 和 Nginx 对比

- Nginx 相对于Apache：
 - 高并发响应性能非常好。（单台万级并发连接30000-50000/s（简单静态页））
 - 反向代理性能非常好。（可用于负载均衡）
 - 内存 和 cpu 占用率低。（为Apache的1/5-1/10）
 - 功能较 Apache 少（常用功能均有）

- 对 php 可使用 cgi 方式和 fastcgi 方式，没有模块编译加载方式。

4、Apache 和 Nginx总结

- Apache 拥有丰富的模块组件支持，稳定性强，BUG少，动态内容处理强。
- Nginx 轻量级，占用资源少，负载均衡，高并发处理强，静态内容处理高效。

中间件介绍

tomcat
weblogic
jboss
php
uwsgi

三、Web 中间件

1、web 中间件简介

- 中间件（英语：Middleware）是提供系统软件和应用软件之间连接的软件，以便于软件各部件之间的沟通。中间件处在操作系统和更高级应用程序之间。他充当的功能是：将应用程序运行环境与操作系统隔离，从而实现应用程序开发者不必为更多系统问题忧虑，而直接关注该应用程序在解决问题上的能力。容器就是中间件的一种。
- 中间件是一类能够为一种或多种应用程序合作互通、资源共享，同时还能够为该应用程序提供相关的服务的软件。（注意：中间件是一类软件的总称，不是单独的一个软件）
- web中间件也叫做web服务器或者web容器
- 正常情况下一切web的访问顺序是：web浏览器—服务器(硬件)—web容器—web应用服务器—数据库服务器。

2、常见的 web 中间件

1、JAVA 中间件

1、Tomcat

Tomcat 是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Java Web 应用服务器（Servlet 容器）。实际上Tomcat 部分是Apache 服务器的扩展，但它是独立运行的，所以当你运行 tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。Tomcat默认使用 8080 号端口

2、Weblogic

WebLogic 是美国Oracle公司出品的一个application server，确切的说是一个基于J2EE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中。Weblogic默认端口是 7001。

3、Jboss

Jboss 是一个基于Java EE的开放源代码的应用服务器。它不但是Servlet容器，而且也是EJB容器，从而受到企业级开发人员的欢迎，从而弥补了Tomcat只是一个Servlet容器的缺憾。JBoss是一个管理EJB的容器和服务。但JBoss核心服务不包括支持 servlet/JSP 的WEB容器，一般与 Tomcat 或 Jetty 绑定使用。Jboss默认端口号是8080。

4、Webshere

WebSphere 是 IBM 的软件平台。它包含了编写、运行和监视全天候的工业强度的按需应变 Web 应用程序和跨平台、跨产品解决方案所需要的整个中间件基础设施，如服务器、服务和工具。WebSphere 提供了可靠、灵活和健壮的软件。WebSphere 是一个模块化的平台，基于业界支持的开放标准。可以通过受信任和持久的接口，将现有资产插入 WebSphere，可以继续扩展环境。WebSphere 可以在许多平台上运行，包括 Intel、Linux 和 z/OS。Webshere默认端口号是 9080。

2、Python 中间件

1、wsgi

一种实现python解析的通用接口标准/协议，跟 wsgi一样概念的有Ruby语言的Rack，可以认为wsgi是一种通用的接口标准或者接口协议，实现了python web程序与服务器之间交互的通用性。有了这个东西，web.py 或者bottle或者django等等的python web开发框架，就可以轻松地部署在不同的web server上了，不需要做任何特殊配置（也需要一些小小的配置调整）

2、uwsgi

uwsgi 同WSGI一样是一种通信协议，是一个 uWSGI 服务器自有的协议，它用于定义传输信息的类型（type of information），每一个uwsgi packet前4byte为传输信息类型描述，它与WSGI相比是两样东西。

3、uWSGI

一种python web server或称为Server/Gateway，uwsgi类似tornadoweb或者flup，是一种python web server，uWSGI是实现了uwsgi和WSGI两种协议的Web服务器，负责响应python 的web请求。因为apache也好，nginx也罢，它们自己都没有解析动态语言如php的功能，而是分派给其他模块来做，比如apache可以说内置了php模块，支持的非常爽，让人感觉好像apache就支持php一样。uwsgi实现了WSGI协议、uwsgi、http等协议。Nginx中HttpUwsgiModule的作用是与uWSGI服务器进行交换。类似的有：flup

3、Php 中间件

1、php-fpm

Php-fpm全称是php fastcgi process manager即php fastcgi进程管理器，相比fastcgi静态的唤起cgi，fpm能根据访问的压力动态的唤起cgi进程和销毁以到达动态的调整cgi数量，这样可以有效的使用内存。除此之外还有其它的一些优点，比如，fpm还可以平滑的重载php配置；由于fpm是使用Unix-Socket来和服务器通讯，所以也不用再配置cgi端口；fpm有更好的状态输出和slowlog日志，502的时候能给出更多的错误细节。

2、CGI

CGI是一个通用的web标准，可以粗略的认为CGI就是一种用任意语言写Web程序的标准。

3、FastCGI

FASTCGI是Web服务器（ex:nginx）和语言解释器(ex:uWsgi)两者底层的通信协议的规范，是对CGI的开放的扩展。

4、Php-FastCGI

- Web Server 启动时载入FastCGI进程管理器（IIS ISAPI或Apache Module）；

- FastCGI进程管理器自身初始化，启动多个CGI解释器进程 (在任务管理器中可见多个php-cgi.exe) 并等待来自Web Server的连接。
- 当客户端请求到达Web Server时，FastCGI进程管理器选择并连接到一个CGI解释器。Web server将CGI环境变量和标准输入发送到FastCGI子进程php-cgi.exe。
- FastCGI子进程完成处理后将标准输出和错误信息从同一连接返回Web Server。当FastCGI子进程关闭连接时，请求便告处理完成。FastCGI子进程接着等待并处理来自FastCGI进程管理器 (运行在WebServer中) 的下一个连接。在正常的CGI模式中，php-cgi.exe在此便退出了。

四、Apache 服务的搭建与配置

1、Apache 服务的搭建

1、系统

操作系统：centos7 (CentOS-7-x86_64-Minimal-1708)

硬件：(这个根据项目运行和配置建议设置，一般我先配个1核1G)

2、关闭 selinux

1、修改 selinux 的配置文件

```
[root@qfedu.com ~]# vim /etc/selinux/config
```

- 将内容中的SELINUX=enforcing修改为SELINUX=disabled

2、关闭 selinux

```
[root@qfedu.com ~]# setenforce 0
[root@qfedu.com ~]# reboot
[root@qfedu.com ~]# sestatus
SELinux status: disabled
```

3、关闭防火墙

```
[root@qfedu.com ~]# systemctl stop firewalld.service
```

4、安装 epel 源

```
[root@qfedu.com ~]# yum -y install epel-release.noarch
```

5、yum 安装 Apache软件

```
[root@qfedu.com ~]# yum -y install httpd*
[root@qfedu.com ~]# rpm -qa | grep httpd      --查看安装的http包
[root@qfedu.com ~]# rpm -qa | grep httpd
httpd-tools-2.4.6-90.el7.centos.x86_64
httpd-itk-2.4.7.04-2.el7.x86_64
httpd-manual-2.4.6-90.el7.centos.noarch
httpd-2.4.6-90.el7.centos.x86_64
httpd-devel-2.4.6-90.el7.centos.x86_64
```

安装成功后，会产生下面两个文件

```
[root@qfedu.com ~]# /etc/httpd/conf/httpd.conf # 主配置文件
[root@qfedu.com ~]# /var/www/html # 默认网站家目录
```

6、认识配置文件里的主要参数

```
[root@qfedu.com ~]# vim /etc/httpd/conf/httpd.conf
serverRoot "/etc/httpd" # 存放配置文件的目录
Listen 80 # Apache服务监听端口
User apache # 子进程的用户
Group apache # 子进程的组
ServerAdmin root@localhost # 设置管理员邮件地址
DocumentRoot "/var/www/html" # 网站家目录

# 设置DocumentRoot指定目录的属性
<Directory "/var/www/html"> # 网站容器开始标识
Options Indexes FollowSymLinks # 找不到主页时，以目录的方式呈现，并允许链接到网站根目录以外
AllowOverride None # none不使用.htaccess控制,all允许
Require all granted # granted表示运行所有访问，denied表示拒绝所有访问
</Directory> # 容器结束
DirectoryIndex index.html # 定义主页文件，当访问到网站目录时如果有定义的主页文件，网站会自动访问
AddDefaultCharset UTF-8 # 字符编码，如果中文的话，有可能需要改为gb2312或者gbk,因你的网站文件的默认编码而异
```

7、启动 Apache 网站

```
[root@qfedu.com ~]# systemctl start httpd.service
```

8、查看httpd服务是否启动

```
[root@qfedu.com ~]# lsof -i:80
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
httpd    20585  root   4u   IPv6  402909      0t0  TCP *:http (LISTEN)
httpd    20586  apache 4u   IPv6  402909      0t0  TCP *:http (LISTEN)
httpd    20587  apache 4u   IPv6  402909      0t0  TCP *:http (LISTEN)
httpd    20588  apache 4u   IPv6  402909      0t0  TCP *:http (LISTEN)
httpd    20589  apache 4u   IPv6  402909      0t0  TCP *:http (LISTEN)
httpd    20590  apache 4u   IPv6  402909      0t0  TCP *:http (LISTEN)
```

9、访问验证 httpd 服务

- 启动成功后使用浏览器或者 curl 输入自己的 IP 地址会看到一个欢迎页面

```
[root@qfedu.com ~]# curl 192.168.1.20
```

- 每次打开浏览器不是很方便，因此我们可以使用文本浏览器，方便测试。

```
[root@qfedu.com ~]# yum -y install elinks # 安装elinks文本浏览器
[root@qfedu.com ~]# elinks 192.168.5.20 # 按ctrl + c退出
```

10、Apache 配置实例

建立网站主页,在网站根目录下建立一个主页文件

格式1 :

```
[root@qfedu.com ~]# echo 'main page' > /var/www/html/index.html      # 往
index.html添加内容
[root@qfedu.com ~]#systemctl restart httpd.service                  # 重启服务
[root@qfedu.com ~]# firefox http://192.168.1.20                    # 在浏览器进行
测试, 显示的信息为刚才我们输入的内容main page
```

格式2 :

```
[root@qfedu.com ~]# vim /var/www/html/index.html                  # 把主页文件写
成html标签的格式, 添加一下内容
<html>
<head>
<title>测试站点</title>
</head>
<body>
<center><h1>欢迎来到测试站点!@_@</h1></center>
</body>
</html>
[root@qfedu.com ~]# systemctl restart httpd.service              # 重启服务
[root@qfedu.com ~]# firefox http://192.168.1.20                  # 在浏览器进行
测试, 显示的信息为刚才我们输入的内容main page
```

例2 : 将网站家目录修改成 : /www目录

```
[root@qfedu.com ~]# mkdir /www                                    # 创建/www目录

[root@qfedu.com ~]# vim /etc/httpd/conf/httpd.conf
119 DocumentRoot "/www"                                          # 修改网站根目录
为/www
131 <Directory "/www">                                           # 把这个也对应的修
改为/www

[root@qfedu.com ~]# systemctl restart httpd.service              # 重新启动apache
服务

[root@qfedu.com ~]# echo "这是新修改的网站家目录/www" > /www/index.html      # 往
index.html添加内容

[root@qfedu.com ~]# firefox http://192.168.1.20                  # 访问网站, 看到新
网站根目录下的刚添加的信息
```

例3 : 修改主页类型或者主页名


```
[root@qfedu.com ~]# vim /etc/httpd/conf/httpd.conf
164 DirectoryIndex index.php # 将index.html改成index.php

[root@qfedu.com ~]# systemctl reload httpd.service # 重新加载服务或重启

[root@qfedu.com ~]# echo 'php main page' > /www/index.php # 这时我们创建一个index.php页面，再使用浏览器访问就能看到了
[root@qfedu.com ~]# elinks 192.168.1.20
```

11、Apache 常用命令

```
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl -M # 查看常见的模块（包括动态和静态）
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl -l # 查看加载的静态模块
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl -t # 检查配置文件有无语法错误
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl graceful # 加载配置文件，但不重启
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl start/restart/stop # 启动/重启/停止apache服务
```

12、Apache 配置用户认证

```
[root@qfedu.com ~]# vim /usr/local/apache2/conf/extra/httpd-vhosts.conf
<Directory /data/www/abc> # /data/www/abc为将要访问的页面的目录
    AllowOverride AuthConfig
    AuthName "cct"
    AuthType Basic
    AuthUserFile /data/.htpasswd # 指定存放用户名和密码的文件
    require valid-user
</Directory>
```

- htpasswd 生成密码文件

```
[root@qfedu.com ~]# htpasswd -c /data/.htpasswd cct # 指定用户名为cct，输入自己设定的密码
[root@qfedu.com ~]# /usr/local/apache2/bin/apachectl graceful 加载配置
```

13、Apache 配置默认虚拟主机

```
[root@qfedu.com ~]# vim /usr/local/apache2/conf/extra/httpd-vhosts.conf
<VirtualHost *:80>
    DocumentRoot "/tmp/111" # 该目录可以不存在
    ServerName 111.com
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "/data/www"
    ServerName www.test.com
    ServerAlias www.aaa.com
</VirtualHost>
```

- 将这段设定放在虚拟主机配置的最上面。除了已作过设置的域名（aaa、test）外，其余访问的域名均跳转到/tmp/111下。

14、Apache 配置 rewrite规则

- Apache 中 rewrite 规则代码均写在 <IfModule mod_rewrite.c> 模块下

1、域名301跳转

web服务可能会用到多个域名，域名有主有次，输入次域名会自动跳转到主域名进行访问。设定为301永久跳转，302是暂时跳转。

```
[root@qfedu.com ~]# vim /usr/local/apache2/conf/extra/httpd-vhosts.conf
<IfModule mod_rewrite.c>
RewriteEngine on      #打开rewrite功能
RewriteCond %{HTTP_HOST} ^www.aaa.com$ [OR]
RewriteCond %{HTTP_HOST} ^www.bbb.com$
RewriteRule ^/(.*)$ http://www.test.com/$1 [R=301,L]
</IfModule>
```

调用了rewrite模块，RewriteCond 是跳转条件，RewriteRule 是跳转规则。

2、禁止指定 user_agent

```
RewriteCond %{HTTP_USER_AGENT} ^.*curl.* [NC,OR] # 禁止curl和chrome浏览器访问
RewriteCond %{HTTP_USER_AGENT} ^.*chrome.* [NC] # NC是不区分大小写的意思
RewriteRule .* - [F] # 为禁止的意思
```

3、通过 rewrite 限制某个目录

```
RewriteCond %{REQUEST_URI} ^.*tmp/.* [NC] # 禁止访问tmp目录
RewriteRule .* - [F]
```

4、rewrite 规则

- R=301 强制外部重定向
- [F]禁用URL,返回403HTTP状态码。
- NC 不区分大小写
- [OR] 或者

5、rewrite 变量

```
%{HTTP_USER_AGENT}      # 表示：访问的user_agent
%{HTTP_HOST}             # 表示：当前访问的网址，只是指前缀部分，格式是www.xxx.com不包括“http://”和“/”
%{REQUEST_URI}           # 表示：表示访问的相对地址，就是相对根目录的地址，就是域名/后面的成分，格式上包括最前面的“/”，
www.123.com/xiang/1.html # www.123.com表示HOST，xiang/1.html表示URI
```

15、Apache 配置日志切割与管理

- web服务器出现大量的访问日志和错误日志，所以我们要对日志进行切割，方便管理人员的查询。一些并不重要的记录，可以规定日志中不显示这些。
- 打开虚拟主机配置文件里的日志记录，日志的路径在/usr/local/apache2/logs下

```
ErrorLog "logs/test.com-error_log"          # 错误日志
CustomLog "logs/test.com-access_log" common  # common 是日志类型
```

- 在 apache 的主配置文件里日志的相关格式规定

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- %h 是来源ip
- %u 访问的user
- %t时间
- %r 动作

- 有些时候一些用户使用代理服务器，这里的%h会显示代理ip，在%h后面加上%{X-FORWARDED-FOR}i 即显示真实ip。

1、日志切割

修改虚拟主机的access log的定义：

```
CustomLog "|/usr/local/apache2/bin/rotatelogs -l
/usr/local/apache2/logs/test.com-access_%Y%m%d_log 86400" combined # %Y%m%d规
定年月日； 86400秒即为一天切割。
```

2、不记录指定文件类型的日志

```
ErrorLog "logs/test.com-error_log"
SetEnvIf Request_URI ".*\.(gif)$" image-request # 对 gif 文件做一个标记 image-
request, 也可以写成 abc
SetEnvIf Request_URI ".*\.(jpg)$" image-request # setenvif 要和 customlog 写在一
起
SetEnvIf Request_URI ".*\.(png)$" image-request
SetEnvIf Request_URI ".*\.(bmp)$" image-request
SetEnvIf Request_URI ".*\.(swf)$" image-request
SetEnvIf Request_URI ".*\.(js)$" image-request
SetEnvIf Request_URI ".*\.(css)$" image-request
CustomLog "|/usr/local/apache2/bin/rotatelogs -l
/usr/local/apache2/logs/test.com-access_%Y%m%d_log 86400" combined env=!image-
request # 不记录做过标记的文件
```

注意：SetEnvIf要和customlog写在一起，中间不要有分隔。

16、配置静态缓存

- 对于图片，html，css，Javascrpts等，让客户端缓存，让访问时不需要去服务器上访问，提高用户体验，需要提前确认是否支持 mod_expires.c模块

```
<IfModule mod_expires.c>
ExpiresActive on
ExpiresByType image/gif "access plus 1 days"
ExpiresByType image/jpeg "access plus 24 hours"
ExpiresByType image/png "access plus 24 hours"
ExpiresByType text/css "now plus 2 hours"
ExpiresByType application/x-javascript "now plus 2 hours"
ExpiresByType application/x-shockwave-flash "now plus 2 hours"
ExpiresDefault "now plus 0 min"
</IfModule>
```

17、配置防盗链

防止其他的网站大量使用自己网站里的一些图片，流量跑的是自己的网站，造成带宽的浪费。防止图片被盗用。

在虚拟主机的配置文件里加上：

```
SetEnvIfNoCase Referer "^http://.*\.test\.com" local_ref # 给网址做标注
<filesmatch "\.(txt|png|gif|doc|mp3|zip|rar|jpg|css|js)"> # 规定一下哪些类型的文件
    Order Allow,Deny # 禁止引用
    Allow from env=local_ref # 允许local_ref引用
</filesmatch>
```

18、访问控制

1、针对某些目录进行访问控制，禁止一些IP地址的访问

```
<Directory "/data/www">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    deny from 192.168.206.1 #该IP地址为禁止访问的ip地址
</Directory>
```

2、针对 URI 做限制访问

```
<filesmatch "(.*)admin(.*)">
    Order allow,deny
    Allow from all
    Deny from 192.168.206.1
</filesmatch>
```

- 所有uri中包括admin的都进行ip访问限制

19、apache 禁止解析 php

- 多用在在网络安全方面，比如：某些目录上传文件，避免上传的有木马，针对某些路径禁止解析php。
- 编辑虚拟主机的配置文件

```
<Directory /data/www/data>      # 对data目录进行php解析限制。
php_admin_flag engine off        # 将php解析引擎关闭
<filesmatch "(.*)php">          # 匹配
order deny,allow
deny from all                    # 禁止解析所有，若不加filesmatch，只是将engine off，
在浏览器访问该文件时，会将php文件下载下来，这样不好
</filesmatch>
</Directory>
```

五、Nginx 服务的搭建与配置

1、系统环境

- OS: CentOS Linux release 7.5.1804 (Core)

2、关闭 selinux

1、修改 selinux 的配置文件

```
[root@qfedu.com ~]# vim /etc/selinux/config
```

- 将内容中的SELINUX=enforcing修改为SELINUX=disabled

2、关闭 selinux

```
[root@qfedu.com ~]# setenforce 0
[root@qfedu.com ~]# reboot
[root@qfedu.com ~]# sestatus
SELinux status: disabled
```

3、关闭防火墙

```
[root@qfedu.com ~]# systemctl stop firewalld.service
```

4、安装 epel 源

```
[root@qfedu.com ~]# yum -y install epel-release.noarch
```

5、设置 nginx 安装源

1、安装官方提供 centos 安装 nginx 源

```
[root@qfedu.com ~]# rpm -Uvh
http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-
0.el7ngx.noarch.rpm
```

2、手动添加 nginx 安装源

```
[root@qfedu.com ~]# vim /etc/yum.repos.d/nginx.repo
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1
```

6、安装 Nginx

- 通过yum search nginx看看是否已经添加源成功。如果成功则执行下列命令安装Nginx。

```
[root@qfedu.com ~]# yum install -y nginx
```

7、启动 Nginx并设置开机自动运行

```
[root@qfedu.com ~]# systemctl start nginx.service
[root@qfedu.com ~]# systemctl enable nginx.service
```

8、配置 Nginx

1、Nginx 的主配置文件

```
[root@qfedu.com ~]# vim /etc/nginx/nginx.conf
```

2、Nginx 配置支持 PHP

- 在 /etc/nginx/conf.d 目录下存放着多个配置文件 这些配置项会在Nginx运行时加载到主配置项目中 (类似虚拟机)
- Nginx是通过php-fpm来通讯的，所以需要监听9000端口。
- 在这个目录下生成一个自己的配置文件例如 admin.conf

```
[root@qfedu.com ~]# vim /etc/nginx/conf.d/admin.conf
```

写入以下内容

```
server {
    listen      80; #端口
    server_name  admin1.test.com admin.test.com ; # 域名
    root        /var/www/card/public; # 网站根目录
    index       index.php index.html index.htm;
    location / {
        if (!-e $request_filename) {
            rewrite ^(.*)$ /index.php?s=/ $1 last; #主要配置隐藏url中index.php
            break;
        }
    }
    location ~ \.php$ {
        #root        /var/www/card/public;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include      fastcgi_params;
    }
}
```

```
}
```

3、Nginx 配置反向代理

```
upstream test{
    server 192.168.68.43:8080 weight=1;
    server 192.168.68.44:8080 weight=1;
}
server {
    listen      80;
    server_name porxy.zxy.com;

    location / {
        proxy_pass # 这里可直接写IP地址进行配置 如果需要配置负载均衡 可以使用
http://test 和upstream名称一致
        # 以下是一些反向代理的配置可删除
        proxy_redirect      off;
        proxy_set_header    Host $host; # 指定请求的服务器的域名和端口号
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for; # 后端的
Web服务器可以通过X-Forwarded-For获取用户真实IP

        client_max_body_size      10m; # 允许客户端请求的最大单文件字节数
        client_body_buffer_size   128k; # 缓冲区代理缓冲用户端请求的最大字节数
        proxy_connect_timeout     300; # nginx跟后端服务器连接超时时间(代理连接
超时)
        proxy_send_timeout       300; # 后端服务器数据回传时间(代理发送超时)
        proxy_read_timeout       300; # 连接成功后, 后端服务器响应时间(代理接收
超时)
        proxy_buffer_size        4k; # 设置代理服务器 (nginx) 保存用户头信息的
缓冲区大小
        proxy_buffers             4 32k; # proxy_buffers缓冲区, 网页平均在32k以
下的话, 这样设置
        proxy_busy_buffers_size   64k; # 高负荷下缓冲大小 (proxy_buffers*2)
        proxy_temp_file_write_size 64k; # 设定缓存文件夹大小, 大于这个值, 将从
upstream服务器传
    }
}
```

五、LA/NMP 架构介绍

- LNMP和LAMP的区别就在于使用Nginx而不是Apache。

1、LNMP 架构

- LNMP是Linux+Nginx+Mysql+PHP的组合方式，其特点是利用Nginx的快速与轻量级，替代以前的LAMP(Linux+Apache+Mysql+PHP)的方式。由于安装方便，并且安装脚本也随时更新。
- LNMP方式的优点：占用VPS资源较少，Nginx配置起来也比较简单，利用fast-cgi的方式动态解析PHP脚本。
- LNMP方式的缺点：php-fpm组件的负载能力有限，在访问量巨大的时候，php-fpm进程容易僵死，容易发生502 bad gateway错误

2、LAMP 架构

- LAMP 即 Linux+Apache+Mysql/MariaDB+Perl/PHP/Python的首字母缩写。这是一组常用来搭建动态网站或者服务器的开源软件。它们本身都是各自独立的程序，但是因为常被放在一起使用，拥有了越来越高的兼容度，共同组成了一个强大的Web应用程序平台。随着开源潮流的蓬勃发展，开放源代码的LAMP已经与J2EE和.Net商业软件形成三足鼎立之势，并且该软件开发的项目在软件方面的投资成本较低，因此受到整个IT界的关注。从网站的流量上来说，70%以上的访问流量是LAMP来提供的，LAMP是最强大的网站解决方案。
- 基于 LAMP 架构设计具有成本低廉、部署灵活、快速开发、安全稳定等特点，是 Web 网络应用和环境的优秀组合。若是服务器配置比较低的个人网站，当然首选 LNMP 架构。

3、LNAMP 架构

- 在大流量的时候。把Apache和Nginx结合起来使用，实现 LNAMP 架构
- LNAMP是Linux+Nginx+Apache+Mysql+PHP的组合方式，其特点是利用Nginx来作为静态脚本的解析，而利用 Nginx的转发特性，将动态脚本的解析转交给Apache来处理，这样，能充分利用两种Web服务器的特点，对于访问量需求较大的站点来说，是一个很不错的选择。
- LNAMP方式的优点：由于Apache本身处理PHP的能力比起php-fpm要强，所以不容易出现类似502 bad gateway的错误。适合访问量较大的站点使用。
- LNAMP方式的缺点：相比LNMP方式会多占用一些资源，另外，配置虚拟主机需要同时修改Nginx和Apache的配置文件，要稍微麻烦一些。

4、LA/NMP 架构各组件介绍

1、Linux

Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

2、Apache

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台和安全性被广泛使用，是最流行的Web服务器端软件之一。它快速、可靠并且可通过简单的API扩展，将Perl/Python等解释器编译到服务器中。

Apache的主程序名叫httpd。

3、Nginx

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，供俄国大型的入口网站及搜索引擎 Rambler（俄文：Рамблер）使用。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好。

4、MySQL

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MariaDB为MySQL的一个分支软件。

5、Php

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。PHP独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

六、MySQL 安装和配置

1、安装 epel 源

```
[root@qfedu.com ~]# yum -y install epel-release.noarch
```

2、安装 MySQL Repository

```
[root@qfedu..com ~]# yum -y install wget          # 安装 wget 下载工具
[root@qfedu..com ~]# wget https://repo.mysql.com//mysql57-community-release-el7-11.noarch.rpm # 下载 mysql 官方 yum 源安装包
[root@qfedu..com ~]# yum -y localinstall mysql57-community-release-el7-11.noarch.rpm          # 安装 mysql 官方 yum 源
```

3、安装 MySQL

需要安装MySQL Server, MySQL client 已经包括在 server 套件内

```
[root@qfedu..com ~]# yum -y install mysql-community-server mysql          # 安装服务端，客户端
[root@qfedu..com ~]# systemctl start mysqld                              # 启动mysql服务
[root@qfedu..com ~]# systemctl enable mysqld                              # 设置 mysql 服务开机启动
[root@qfedu..com ~]# ls /var/lib/mysql                                    # 查看 mysql 安装
[root@qfedu..com ~]# grep 'tqfeduorary password' /var/log/mysqld.log      # 获取首次登录密码
[root@qfedu..com ~]# mysql -uroot -p'awm3>!QF16zR'                        # 登录mysql数据库
mysql > alter user 'root'@'localhost' identified by 'Qf.123com';          # 修改 mysql 数据库密码（密码必须符合复杂性要求，包含字母大小写，数字，特殊符号，长度不少于8位）
[root@qfedu..com ~]# mysql -uroot -p'Qf.123com'                          # 用新密码登录数据库
```

4、重启 MySQL

```
[root@qfedu..com ~]# /usr/bin/systemctl restart mysqld
```

5、创建 qfedu 库并设置权限

```
[root@qfedu..com ~]# mysql -u root -p
Enter password:
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
```

```

Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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>CREATE DATABASE qfedudb CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 1 row affected (0.00 sec)

mysql>GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX on qfedudb.* TO
'qfedu'@'localhost' IDENTIFIED BY 'Yangge.123com';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'qfedu'@'localhost';
+-----+
+-----+
| Grants for qfedu@localhost |
+-----+
+-----+
| GRANT USAGE ON *.* TO 'qfedu'@'localhost' IDENTIFIED BY PASSWORD
'*841E9705B9F4BD3195B7314CA58A7E3B3B349F71' |
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON
`qfedudb`.* TO 'qfedu'@'localhost' |
+-----+
+-----+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR 'qfedu'@'172.16.0.122';
+-----+
+-----+
| Grants for qfedu@172.16.0.122 |
+-----+
+-----+
| GRANT USAGE ON *.* TO 'qfedu'@'172.16.0.122' IDENTIFIED BY PASSWORD
'*841E9705B9F4BD3195B7314CA58A7E3B3B349F71' |
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON
`qfedudb`.* TO 'qfedu'@'172.16.0.122' |
+-----+
+-----+
2 rows in set (0.01 sec)

mysql> \q
Bye

```

七、MariaDB 安装和配置

- MariaDB数据库管理系统是MySQL的一个分支，主要由开源社区在维护，采用GPL授权许可。

- 开发这个分支的原因之一是：甲骨文公司收购了MySQL后，有将MySQL闭源的潜在风险，因此社区采用分支的方式来避开这个风险。
- MariaDB的目的是完全兼容MySQL，包括API和命令行，使之能轻松成为MySQL的替代品。

1、MariaDB 源配置

```
[root@qfedu..com ~]# vim /etc/yum.repo.d/mariadb.repo
[mariadb]
name = MariaDB
baseurl = https://mirrors.ustc.edu.cn/mariadb/yum/10.4/centos7-amd64
gpgkey=https://mirrors.ustc.edu.cn/mariadb/yum/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

2、MariaDB 安装

```
[root@qfedu..com ~]# yum -y install MariaDB-server MariaDB-client
```

3、MariaDB 基本配置

1、MariaDB 管理命令

```
[root@qfedu..com ~]# systemctl start mariadb      # 启动MariaDB
[root@qfedu..com ~]# systemctl stop mariadb       # 停止MariaDB
[root@qfedu..com ~]# systemctl restart mariadb    # 重启MariaDB
[root@qfedu..com ~]# systemctl enable mariadb     # 设置开机启动
```

2、MariaDB 进程查看

```
[root@qfedu..com ~]# netstat -ntlp |grep 3306
tcp        0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN     3931/mysqld
```

3、MariaDB 数据库初始化

- 为了确保数据库的安全性和正常运转，需要先对数据库程序进行初始化操作。这个初始化操作涉及下面 5 个步骤。
 - 设置 root 管理员在数据库中的密码值(注意，该密码并非 root 管理员在系统中的密码，这里的密码值默认应该为空，可直接按回车键)。
 - 设置 root 管理员在数据库中的专有密码。
 - 随后删除匿名账户，并使用 root 管理员从远程登录数据库，以确保数据库上运行的业务的安全性。
 - 删除默认的测试数据库，取消测试数据库的一系列访问权限。
 - 刷新授权列表，让初始化的设定立即生效。

1、初始化命令

```
[root@qfedu..com ~]# mysql_secure_installation
Enter current password for root (enter for none):          # 初次运行直接回车
设置密码
Set root password? [Y/n] y                                # 是否设置root用户密码，输入y并回车或直接回车
New password:                                              # 设置root用户的密码
Re-enter new password:                                     # 再输入一次你设置的密码
Remove anonymous users? [Y/n] y                            # 是否删除匿名用户，回车
Disallow root login remotely? [Y/n] n                     # 是否禁止root远程登录，回车，不过一般为y
Remove test database and access to it? [Y/n] y            # 是否删除test数据库，回车
Reload privilege tables now? [Y/n] y                      # 是否重新加载权限表，回车
```

2、初始化测试登录

```
[root@qfedu..com ~]# mysql -uroot -p                    # 进入数据库(数据库中的操作命令和mysql是一样的)
```

4、MariaDB 设置utf8编码

```
[root@qfedu..com ~]# vim /etc/my.cnf
[mysqld]
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
skip-character-set-client-handshake
```

4、MariaDB 的使用

- MariaDB 的使用和mysql的语句完全是一样的

1、基本使用

```
# 修改mysql密码
MariaDB [(none)]> set password = PASSWORD('hsz123');
Query OK, 0 rows affected (0.34 sec)

# 创建tests数据库 如果加charset=utf8 表示指定utf8编码
MariaDB [(none)]> create database test;
Query OK, 1 row affected (0.00 sec)

# 进入test数据库
MariaDB [(none)]> use test;
Database changed

# 创建mytest数据表
MariaDB [test]> create table mytest(id int,name char(32));
Query OK, 0 rows affected (0.02 sec)

# 查看数据表
MariaDB [test]> show tables;
+-----+
| Tables_in_test |
+-----+
```

```
| mytest |
+-----+
1 row in set (0.00 sec)

# 查看mytest数据表的表结构
MariaDB [test]> desc mytest;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | YES  |     | NULL    |       |
| name  | char(32)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.05 sec)
```

2、简单的增删改查

```
# 给表增加两条数据
MariaDB [test]> insert into mytest(id,name) values(1,"zero"),(2,"one");
Query OK, 2 rows affected (0.35 sec)
Records: 2  Duplicates: 0  Warnings: 0

# 查看id,name 字段mytest的数据
MariaDB [test]> select id,name from mytest;
+-----+-----+
| id  | name |
+-----+-----+
| 1  | zero |
| 2  | one  |
+-----+-----+
2 rows in set (0.00 sec)

# 删除mytest表中id=3 的数据
MariaDB [test]> delete from mytest id=2;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near
'id=2' at line 1
MariaDB [test]> delete from mytest where id=2;
Query OK, 1 row affected (0.00 sec)

# 查看表的所有数据
MariaDB [test]> select * from mytest;
+-----+-----+
| id  | name |
+-----+-----+
| 1  | zero |
+-----+-----+
1 row in set (0.01 sec)

# 更新表id=1表的字段name=ten
MariaDB [test]> update mytest set name=ten where id=1;
ERROR 1054 (42S22): Unknown column 'ten' in 'field list'
MariaDB [test]> update mytest set name="ten" where id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
MariaDB [test]> select * from mytest;
+-----+-----+
| id  | name |
+-----+-----+
```

```
| 1 | ten |
+-----+-----+
1 row in set (0.01 sec)
```

3、关于用户及权限常用命令

创建用户和密码

```
MariaDB [test]> create user zero@'%' identified by 'zero';
Query OK, 0 rows affected (0.01 sec)
```

mysql 使用 grant 命令对账户进行授权, grant 命令常见格式如下

# grant 权限 on 数据库.表名 to 账户@主机名	对特定数据库中的特定表授权
# grant 权限 on 数据库.* to 账户@主机名	对特定数据库中的所有表给与授权
# grant 权限1,权限2,权限3 on *.* to 账户@主机名	对所有库中的所有表给与多个授权
# grant all privileges on *.* to 账户@主机名	对所有库和所有表授权所有权限

授予用户最大的权限, 所有的权限

```
grant all privileges on *.* to username@'%' identified by 'password';
```

授予zero用户, 只有创建test数据库的权限

```
MariaDB [test]> grant create on test.* to zero@'%' identified by 'zero';
Query OK, 0 rows affected (0.00 sec)
```

所以查询zero用户的数据库只有如下所示

```
[root@qfedu.com ~]# mysql -uzero -pzero
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 5.5.60-MariaDB MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| test |
+-----+
2 rows in set (0.00 sec)
MariaDB [(none)]>
```

授予one创建的权限, 对于所有的库表生效

```
MariaDB [test]> grant create on *.* to one@'%' identified by 'one';
Query OK, 0 rows affected (0.00 sec)
```

所以查询数据库可以显示如下所示

```
[root@qfedu.com ~]# mysql -uone -pone
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 5.5.60-MariaDB MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
```

```
| information_schema |
| mysql              |
| performance_schema |
| test               |
| text               |
+-----+
```

5 rows in set (0.00 sec)

MariaDB [(none)]>

删除one用户

MariaDB [test]> drop user one;

Query OK, 0 rows affected (0.00 sec)

刷新权限

MariaDB [test]> flush privileges;

Query OK, 0 rows affected (0.00 sec)

4、数据库备份与恢复

备份所有数据库命令

[root@qfedu.com ~]# mysqldump -u root -p --all-databases > /tmp/db.dump

Enter password:

[root@qfedu.com ~]# ll /tmp/db.dump

-rw-r--r--. 1 root root 515562 Sep 13 23:00 /tmp/db.dump

备份单个数据库命令

[root@qfedu.com ~]# mysqldump -u root -p text > /tmp/text.sql

Enter password:

[root@qfedu.com ~]# ll /tmp/text.sql

-rw-r--r--. 1 root root 1261 Sep 13 23:01 /tmp/text.sql

将备份的数据库导入

[root@qfedu.com ~]# mysql -uroot -p text2< /tmp/text.sql

删除数据库

MariaDB [text2]> show databases;

```
+-----+
| Database |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| test           |
| text           |
| text2          |
+-----+
```

6 rows in set (0.00 sec)

MariaDB [text2]> drop database text;

Query OK, 0 rows affected (0.00 sec)

MariaDB [text2]> show databases;

```
+-----+
```

Database
information_schema
mysql
performance_schema
test
text2

八、PHP 安装和配置

1、安装 EPEL 源

```
[root@qfedu..com ~]# install epel-release
```

2、安装 REMI 源：

```
[root@qfedu..com ~]# yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

3、安装 Yum 源管理工具：

```
[root@qfedu..com ~]# yum install yum-utils
```

4、安装 PHP7.3

```
[root@qfedu..com ~]# yum install -y php73-php-fpm php73-php-cli php73-php-bcmath  
php73-php-gd php73-php-json php73-php-mbstring php73-php-mcrypt php73-php-  
mysqlnd php73-php-opcache php73-php-pdo php73-php-pecl-crypto php73-php-pecl-  
mcrypt php73-php-pecl-geoip php73-php-recode php73-php-snmp php73-php-soap  
php73-php-xmll
```

- 安装完成后最好重启一下计算机，不然有可能无法启动php

5、启动 php 服务

```
[root@qfedu..com ~]# systemctl enable php73-php-fpm
[root@qfedu..com ~]# systemctl start php73-php-fpm
```

6、设置 PHP

1、查找安装包

```
[root@qfedu..com ~]# rpm -qa | grep 'php'
php73-php-cli-7.3.6-1.el7.remi.x86_64
php73-php-mbstring-7.3.6-1.el7.remi.x86_64
php73-php-pecl-mcrypt-1.0.2-1.el7.remi.x86_64
php73-runtime-2.0-1.el7.remi.x86_64
php73-php-pdo-7.3.6-1.el7.remi.x86_64
php73-php-bcmath-7.3.6-1.el7.remi.x86_64
```



```
php73-php-fpm-7.3.6-1.el7.remi.x86_64
php73-php-mysqlnd-7.3.6-1.el7.remi.x86_64
php73-php-snmp-7.3.6-1.el7.remi.x86_64
php73-php-gd-7.3.6-1.el7.remi.x86_64
php73-php-json-7.3.6-1.el7.remi.x86_64
php73-php-soap-7.3.6-1.el7.remi.x86_64
php73-php-recode-7.3.6-1.el7.remi.x86_64
php73-php-pec1-crypto-0.3.1-5.el7.remi.x86_64
php73-php-common-7.3.6-1.el7.remi.x86_64
php73-php-opcache-7.3.6-1.el7.remi.x86_64
php73-php-pec1-geoip-1.1.1-6.el7.remi.x86_64
```

2、查看 php 安装位置

```
[root@qfedu..com ~]# rpm -ql php73-php-fpm-7.3.6-1.el7.remi.x86_64
/etc/logrotate.d/php73-php-fpm
/etc/opt/remi/php73/php-fpm.conf
/etc/opt/remi/php73/php-fpm.d
/etc/opt/remi/php73/php-fpm.d/www.conf
/etc/opt/remi/php73/sysconfig/php-fpm
/etc/systemd/system/php73-php-fpm.service.d
/opt/remi/php73/root/usr/sbin/php-fpm
/opt/remi/php73/root/usr/share/doc/php73-php-fpm-7.3.6
/opt/remi/php73/root/usr/share/doc/php73-php-fpm-7.3.6/php-fpm.conf.default
/opt/remi/php73/root/usr/share/doc/php73-php-fpm-7.3.6/www.conf.default
/opt/remi/php73/root/usr/share/fpm
/opt/remi/php73/root/usr/share/fpm/status.html
/opt/remi/php73/root/usr/share/licenses/php73-php-fpm-7.3.6
/opt/remi/php73/root/usr/share/licenses/php73-php-fpm-7.3.6/fpm_LICENSE
/opt/remi/php73/root/usr/share/man/man8/php-fpm.8.gz
/usr/lib/systemd/system/php73-php-fpm.service
/var/opt/remi/php73/lib/php/opcache
/var/opt/remi/php73/lib/php/session
/var/opt/remi/php73/lib/php/wddxcache
/var/opt/remi/php73/log/php-fpm
/var/opt/remi/php73/run/php-fpm
```

3、查找php 配置文件位置

```
[root@qfedu..com ~]# find /etc/opt/remi/php73 -name php.ini
/etc/opt/remi/php73/php.ini
```

4、设置快捷命令

```
[root@qfedu..com ~]# sed -i 's/;cgi.fix_pathinfo=1/cgi.fix_pathinfo=0/'
/etc/opt/remi/php73/php.ini
[root@qfedu..com ~]# systemctl restart php73-php-fpm
```

5、php 管理命令

```
[root@qfedu..com ~]# systemctl restart php73-php-fpm      # 重启
[root@qfedu..com ~]# systemctl start php73-php-fpm        # 启动
[root@qfedu..com ~]# systemctl stop php73-php-fpm         # 关闭
[root@qfedu..com ~]# systemctl status php73-php-fpm       # 检查状态
```

6、查看 php 版本

```
[root@qfedu..com ~]# php73 -v
PHP 7.3.0-1+(cli) (built: Dec 6 2018 20:24:55) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.0-dev, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.0-1+ubuntu18.04.1+deb.sury.org+1, Copyright (c) 1999-
2018, by Zend Technologies
```

7、更新 php

- 运行下面的命令系统就会更新所有可以更新的软件包括 PHP

```
[root@qfedu..com ~]# yum update
```

8、安装更多组件

- 上面的一条命令安装 PHP 只是安装了部分 PHP 拓展，更多的软件可见：

```
[root@qfedu..com ~]# yum search php73
Updating Subscription Management repositories.
Last metadata expiration check: 0:27:54 ago on Wed 15 May 2019 10:39:52 AM EDT.
=====
===== Name Exactly Matched: php73
=====
php73.x86_64 : Package that installs PHP 7.3
php73.x86_64 : Package that installs PHP 7.3
=====
===== Name & Summary Matched: php73
=====
php73-syspaths.x86_64 : System-wide wrappers for the php73 package
php73-syspaths.x86_64 : System-wide wrappers for the php73 package
php73-scldevel.x86_64 : Package shipping development files for php73
php73-scldevel.x86_64 : Package shipping development files for php73
php73-php-zstd-devel.x86_64 : php73-php-zstd developer files (header)
php73-runtime.x86_64 : Package that handles php73 Software Collection.
php73-runtime.x86_64 : Package that handles php73 Software Collection.
php73-runtime.x86_64 : Package that handles php73 Software Collection.
php73-php-pec1-psr-devel.x86_64 : php73-php-pec1-psr developer files (header)
php73-php-pec1-psr-devel.x86_64 : php73-php-pec1-psr developer files (header)
php73-php-pec1-raphf-devel.x86_64 : php73-php-pec1-raphf developer files
(header)
php73-php-pec1-raphf-devel.x86_64 : php73-php-pec1-raphf developer files
(header)
php73-php-pec1-propro-devel.x86_64 : php73-php-pec1-propro developer files
(header)
php73-php-pec1-yaconf-devel.x86_64 : php73-php-pec1-yaconf developer files
(header)
php73-php-pec1-propro-devel.x86_64 : php73-php-pec1-propro developer files
(header)
php73-php-pec1-yaconf-devel.x86_64 : php73-php-pec1-yaconf developer files
(header)
php73-php-pec1-xmlDIFF-devel.x86_64 : php73-php-pec1-xmlDIFF developer files
(header)
```

```
php73-php-pec1-swoole4-devel.x86_64 : php73-php-pec1-swoole4 developer files
(header)
php73-php-pec1-xml-diff-devel.x86_64 : php73-php-pec1-xml-diff developer files
(header)
php73-php-zephir-parser-devel.x86_64 : php73-php-zephir-parser developer files
(headers)
php73-php-zephir-parser-devel.x86_64 : php73-php-zephir-parser developer files
(headers)
php73-php-pec1-handlebars-devel.x86_64 : php73-php-pec1-handlebars developer
files (header)
```

```
=====
===== Name Matched: php73
=====
```

```
=====
php73-php.x86_64 : PHP scripting language for creating dynamic web sites
php73-php.x86_64 : PHP scripting language for creating dynamic web sites
php73-build.x86_64 : Package shipping basic build configuration
.....
```

九、基于 LNMP 快速简单搭建 wordpress 平台

1、WordPress 简介

WordPress 是一种使用 PHP 语言开发的博客平台，用户可以在支持 PHP 和 MySQL 数据库的服务器上架设属于自己的网站。也可以把 WordPress 当作一个内容管理系统（CMS）来使用。WordPress 是一款个人博客系统，并逐步演化成一款内容管理系统软件，它是使用 PHP 语言和 MySQL 数据库开发的。用户可以在支持 PHP 和 MySQL 数据库的服务器上使用自己的博客。WordPress 有许多第三方开发的免费模板，安装方式简单易用。不过要做一个自己的模板，则需要你有一定的专业知识。比如你至少要懂的标准通用标记语言下的一个应用 HTML 代码、CSS、PHP 等相关知识。WordPress 官方支持中文版，同时有爱好者开发的第三方中文语言包，如 woprus 中文语言包。WordPress 拥有成千上万个各式插件和不计其数的主题模板样式。

2、基础环境要求

- Mysql
- PHP
- nginx

3、安装 mysql 环境

1、安装mysql

```
[root@qfedu..com ~]# wget -i -c http://dev.mysql.com/get/mysql57-community-
release-e17-10.noarch.rpm
[root@qfedu..com ~]# yum -y install mysql57-community-release-e17-10.noarch.rpm
[root@qfedu..com ~]# yum -y install mysql-community-server
```

2、启动mysql

```
[root@qfedu..com ~]# systemctl start mysqld
```

3、获取默认密码

```
[root@qfedu..com ~]# grep "password" /var/log/mysqld.log
```

4、登录mysql

```
[root@qfedu..com ~]# mysql -uroot -p
```

5、修改密码

```
mysql> alter user 'root'@'localhost' identified by 'Lile@5201314';
```

6、创建wordpress数据库并授权

```
mysql> create database wordpress;
Query OK, 1 row affected (0.00 sec)

mysql> create user 'wordpress'@'localhost' identified by 'Qfedu.123com';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on wordpress.* to 'wordpress'@'localhost';
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

4、安装 nginx 环境

1、安装 nginx

```
[root@qfedu..com ~]# yum install nginx -y
```

2、添加配置文件

```
[root@qfedu..com ~]# vim /etc/nginx/conf.d/wordpress.conf
server {
    listen 80;
    server_name opswordpress.transspay.net;
    root /wordpress;
    location / {
        index index.php index.html index.htm;
        try_files $uri $uri/ /index.php index.php;
    }
    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    location ~ .php$ {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

3、重新加载 nginx 配置

```
[root@qfedu..com ~]# nginx -s reload
```

4、安装 php 环境

1、安装 php-fpm

```
[root@qfedu..com ~]# yum install php-fpm php-mysql -y
```

2、启动 php-fpm

```
[root@qfedu..com ~]# systemctl start php-fpm.service
```

3、检测 php-fpm 端口

- 9000为php-fpm的默认端口

```
[root@qfedu..com ~]# lsof -i:9000
```

5、安装 wordpress

1、下载解压

```
[root@qfedu..com ~]# wget https://wordpress.org/latest.tar.gz  
[root@qfedu..com ~]# tar xf latest.tar.gz /
```

2、设置 wordpress

- 设置wp-config.php文件，根据自己的数据库修改相关的配置

```
[root@qfedu..com ~]# cp /wordpress/wp-config-sample.php /wordpress/wp-config.php  
[root@qfedu..com ~]# vim /wordpress/wp-config.php  
define( 'DB_NAME', 'wordpress' );  
/** MySQL database username */  
define( 'DB_USER', 'wordpress' );  
/** MySQL database password */  
define( 'DB_PASSWORD', 'Qfedu.123com' );  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );  
/** Database Charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8' );  
/** The Database Collate type. Don't change this if in doubt. */  
define( 'DB_COLLATE', '' );
```

3、设置密钥的部分

- 可以在官网获取：<https://api.wordpress.org/secret-key/1.1/salt/>

```
define('AUTH_KEY',
'h#f>^]S]8_%v|c5y8$r{;OOQ1SWuFV4`1Tk^g=MimlUQ`z{x(;^aMo)yFvm6}zJ;');
define('SECURE_AUTH_KEY', 'h+Qs$r|5t[^I,}H[$s/~n+)Jt4#<{=D|@RtZU.LJa pX
dupeRrUE4Cra-^Bhb)2');
define('LOGGED_IN_KEY', 'zbZEh/#7wZYk_Bz4>t}W~L{s8V$!0qL1/*
{z^twj6A}~!3i1I!9Iv`~L>|0z[J');
define('NONCE_KEY', 'ou7nI)_EhxqlT6S)+o/?cF5koyA{c&<n++(--
hpor<}jU&s#OnHD`M>az/tMU2[');
define('AUTH_SALT', 'mw7Q&w|wblshXnZ{b rj/vk03%_<:4Zc$~3.&04F[S__$_e-w-
6!SZ^J!}[e-u&');
define('SECURE_AUTH_SALT', 'r-flzGU2Uu>(sfL?F`_]giywni%ukt}iym|)wuH^rFom~Q
:Lw+m8m}IP9ks0F{[');
define('LOGGED_IN_SALT', 'Ox@>-u+@lgS0#
(MQyL:Xan|&+s.Gq[$c+:sGG&u]-51T!d]Z}:%*aG7H!6K$;>@7');
define('NONCE_SALT', 'j*dbFR^.;ng_Fm5</5#rmk5/$VngTERa$-
D0Vdv#@jow-];abKrT,_p@<_/gne}w');
```

```
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', '/wordpress' );
}
```

4、访问配置 Wordpress

1、打开浏览器输入IP/域名，

- 弹出安装界面。填上相关网站信息，点击“安装Wordpress”按钮即可完成。

千锋云计算学院



欢迎

欢迎使用著名的WordPress五分钟安装程序！请简单地填写下面的表格，来开始使用这个世界上最具扩展性、最强大的个人信息发布平台。

需要信息

您需要填写一些基本信息。无需担心填错，这些信息以后可以再次修改。

站点标题

用户名

用户名只能含有字母、数字、空格、下划线、连字符、句号和“@”符号。

密码

强

隐藏

重要：您将需要此密码来登录，请将其保存在安全的位置。

您的电子邮件

请仔细检查电子邮件地址后再继续。

对搜索引擎的可见性

☐ 建议搜索引擎不索引本站点

搜索引擎将本着自觉自愿的原则对待WordPress提出的请求。并不是所有搜索引擎都会遵守这类请求。

安装WordPress

2、安装成功界面



成功！

WordPress安装完成。谢谢！

用户名

admin

密码

您设定的密码。

登录

3、安装完成后登录

世界，您好！

欢迎使用WordPress。这是您的第一篇文章。编辑或删除它，然后开始写作吧！

👤 banana 🕒 2019年2月10日 📁 未分类 💬 有1条评论 ✎ 编辑

5、使用默认模板的网站前端

- **注：**在安装主题时，会需要你安装ftp，我们不需要安装，直接在wp-config.php配置文件里添加如下几行即可

```
define("FS_METHOD","direct");  
define("FS_CHMOD_DIR", 0777);  
define("FS_CHMOD_FILE", 0777);
```

千锋云计算学院