

# Linux服务器构建与运维管理

## 第03章：网站服务器

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

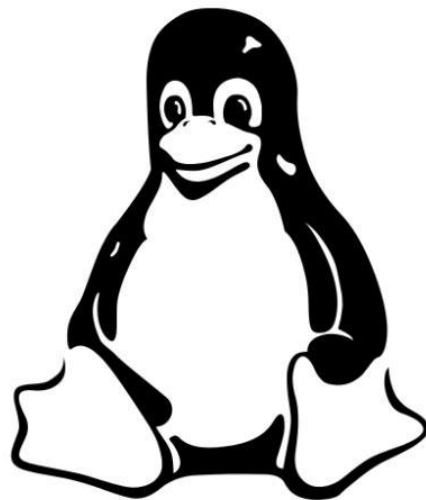
<http://linux.xg.hactcm.edu.cn>  
<http://www.51xueweb.cn>

河南中医药大学信息管理与信息系统教研室  
信息技术学院网络与信息系统科研工作室

2021.3

# 提纲

- Web服务器
- Apache服务器
- 使用Apache发布网站
  - 使用Apache发布静态网站
  - 使用Apache发布多个网站
- LAMP
  - 基于CentOS实现LAMP
  - 使用WordPress实现内容网站服务
  - 监控Apache的性能
- 提升Apache的安全性
- 通过Apache日志分析洞察网站业务



# 1.Web服务器

## 1.1 Web服务器简介

- Web服务器是一种驻留于因特网上的计算机程序，实现两个作用：
  - 放置网站程序，让用户通过浏览器访问
  - 放置数据文件，让用户下载
  
- Web服务器端对客户端提供的网站程序有两种类型：
  - 静态文档：html、JS
  - 动态文档：PHP、ASP.net、JSP、Ruby、Python
  
- Web服务器使用HTTP进行通信，因此Web服务器亦称为HTTP服务器。



# 1.Web服务器

## 1.1 Web服务器简介

- 由于不同的Web服务器对HTTP请求的处理方式并不完全相同，进行相应的处理时采用的资源分配策略和调度的方式也各有差异。
- 随着技术的发展，适应为各种不同功能、不同环境的Web服务器不断出现，目前常用的Web服务器是Apache、IIS、Tomcat、Nginx等。
- Apache是Linux平台中应用最广泛的Web服务器。



# 1.Web服务器

## 1.2 Web服务器工作原理

- Web服务器的处理流程一般分为四步：连接过程、请求过程、应答过程以及关闭连接。
  - 连接过程：
    - 根据输入的URL地址，Web客户端（如浏览器）连接到相应的Web服务器上。
  - 请求过程：
    - Web客户端运用socket向其服务器提出各种请求。
  - 应答过程：
    - Web服务器接收到请求后实施任务处理，使用HTTP协议把任务处理的结果传输到Web客户端。Web客户端获取数据后展示给用户。
  - 关闭连接：
    - 当应答过程完成以后，Web服务器和Web客户端之间断开连接。



# 1.Web服务器

## 1.2 Web服务器工作原理

- Web服务器的处理流程一般分为四步：连接过程、请求过程、应答过程以及关闭连接。



## 2.Apache服务器

### 2.1 Apache简介

- Apache是最常用的开源网站服务器软件之一，支持UNIX、Linux、Windows等操作系统。
  - Apache官网为<https://www.apache.org>
  - Apache目前主流版本为2.4.37





### COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

- Projects
- People
- Community
- License
- Sponsors

## THE WORLD'S LARGEST OPEN SOURCE FOUNDATION

- All volunteer community
- 227M+ lines of code in stewardship
- 4.2B+ lines of code changed
- 3.9M+ code commits
- 813 individual ASF Members
- 8,000+ Apache Committers
- 39,000+ code contributors
- 460,000+ people involved in our communities
- 350+ Projects and Initiatives
- 300+ Top-Level Projects
- 45 podlings in the Apache Incubator
- ~2 Petabytes source code downloads from Apache mirrors
- 24M+ emails across 1,400+ mailing lists
- Web requests received from every Internet-connected country on the planet
- 35M+ weekly page views across apache.org

**\$20B+ worth of Apache Open Source software products are made available to the public-at-large at 100% no cost, and benefit billions of users around the world.**

**[Make a one-time or recurring donation](#) | [Corporate Support and ASF Sponsorship](#)**







## COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"



- Projects
- People
- Community
- License
- Sponsors

The mission of the Apache Software Foundation (ASF) is to provide software for the public good. We do this by providing services and support for many like-minded software project communities consisting of individuals who choose to participate in ASF activities.

### WHAT IS THE ASF?

Established in 1999, the ASF is a US 501(c)(3) charitable organization, funded by individual donations and corporate sponsors. Our all-volunteer board oversees more than 350 leading Open Source projects, including Apache HTTP Server -- the world's most popular Web server software.

The ASF provides an established framework for intellectual property and financial contributions that simultaneously limits potential legal exposure for our project committers. Through the ASF's meritocratic process known as "The Apache Way," more than 730 individual Members and 7,000 Committers successfully collaborate to develop freely available enterprise-grade software, benefiting millions of users worldwide: thousands of software solutions are distributed under the Apache License; and the community actively participates in ASF mailing lists, mentoring initiatives, and ApacheCon, the Foundation's official user conference, trainings, and expo.

### HOW DID THE ASF AND APACHE® PROJECTS GROW?

Formerly known as the Apache Group, the ASF was incorporated in 1999 as a membership-based, not-for-profit corporation in order to ensure that the Apache projects continue to exist beyond the participation of individual volunteers. Individuals who have demonstrated a commitment to collaborative open-source software development, through sustained participation and contributions within the Foundation's projects, are eligible for membership in the ASF. An individual is awarded membership after nomination and approval by a majority of the existing ASF members. Thus, the ASF is governed by the community it most directly serves -- the people collaborating within its projects.

### HOW ARE THE ASF AND APACHE PROJECTS GOVERNED?

The ASF members periodically elect a Board of Directors to manage the organizational affairs of the Foundation, as accorded by the ASF Bylaws. The Board, in turn, appoints a number of officers to oversee the day-to-day operations of the Foundation. A number of public records of our operation are made available to the community. A more detailed explanation of How the ASF works in terms of day to day operations is available, and the Apache Community Development project's goal is to help newcomers learn more about the Apache Software Foundation.

Individual Apache projects are in turn governed directly by Project Management Committees (PMC) made up of individuals who have shown merit and leadership within those projects. There are detailed descriptions of ASF and project governance models.

### WHAT IS THE ROLE OF ASF MEMBERS?

## APACHE PROJECT LIST

Overview

All Projects

### BY CATEGORY

Attic  
Big Data  
Build Management  
Cloud  
Content  
Databases  
FTP  
Graphics  
HTTP  
HTTP-module  
Incubating  
JavaEE  
Libraries  
Mail  
Mobile  
Network-client  
Network-server  
OSGi  
RegExp  
Retired  
Search  
Security

### BY NAME

HTTP Server

A  
Accumulo  
ActiveMQ  
Airavata  
Airflow  
Allura  
Ambari  
Ant  
Any23  
APISIX  
Archiva  
Aries  
Arrow  
AsterixDB  
Atlas  
Attic  
Avro  
Axis  
Bahir  
Beam  
Bigtop  
Bloodhound

Commons  
Community  
Development  
Cordova  
CouchDB  
Creadur  
cTAKES  
Curator  
CXF  
D  
Daffodil  
DataFu  
DataSketches  
DB  
DeltaSpike  
Directory  
Drill  
Druid  
Dubbo  
E  
ECharts  
Empire-db  
F

Gump  
H  
Hadoop  
HAWQ  
HBase  
Helix  
Hive  
HttpComponents  
Hudi  
I  
Iceberg  
Ignite  
Impala  
Incubator  
IoTDB  
Isis  
J  
Jackrabbit  
James  
jclouds  
Jena  
JMeter  
Johnzon  
Joshua

M  
MADlib  
Mahout  
ManifoldCF  
Maven  
Mesos  
MetaModel  
MINA  
Mnemonic  
MyFaces  
Mynewt  
N  
NetBeans  
NiFi  
Nutch  
O  
OFBiz  
Olingo  
OODT  
Oozie  
OpenJPA  
OpenMeetings  
OpenNLP  
OpenOffice

Portals  
Pulsar  
Q  
Qpid  
R  
Ranger  
Ratis  
REEF  
River  
RocketMQ  
Roller  
Royale  
Rya  
S  
Samza  
Santuario  
Serf  
ServiceComb  
ServiceMix  
ShardingSphere  
Shiro  
SINGA  
SIS  
SkyWalking

T  
Tapestry  
Tcl  
Tez  
Thrift  
Tika  
TinkerPop  
Tomcat  
TomEE  
Traffic Control  
Traffic Server  
Trafodion  
Turbine  
TVM  
U  
UIMA  
Unomi  
Usergrid  
V  
VCL  
Velocity  
W  
Web Services  
Whimsy

Apache Software Foundation, ASF: Apache软件基金会

ASF早期发布了一款Web服务器软件叫做Apache, 取得了巨大的成功。  
ASF将Apache Web服务器软件自2.0版本起, 更名为HTTP Server。  
Apache服务器软件是准确名称是: Apache HTTP Server。  
使用Apache服务器软件仅是习惯的说法而已。

Cocoon

Groovy  
Guacamole

Lucene.Net

Portable Runtime  
(APR)

SystemDS

COMMUNITY

Overview

INNOVATION

Incubator

TECH OPERATIONS

Developer Information

PRESS

Overview

LEGAL

Legal Affairs

## 2.Apache服务器

### 2.2 Apache的特性

#### □ Apache的主要特性：

- 支持最新的HTTP协议和多种方式的HTTP认证
- 支持基于文件的配置
- 支持基于IP和域名的虚拟网站配置
- 支持通用网关接口，支持PHP、FastCGI、Perl、JavaServlets等
- 支持服务器状态监控
- 支持服务器日志记录和日志格式自定义设置
- 支持服务器端包含指令（SSI）
- 支持安全Socket层（SSL）
- 集成代理服务器模块
- 提供用户会话过程的跟踪
- 支持第三方软件提供的功能模块，实现灵活扩展





### The Number One HTTP Server On The Internet

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 25th birthday as a project in February 2020.

The Apache HTTP Server is a project of [The Apache Software Foundation](#).

### Apache httpd 2.4.46 Released

2020-08-07

The Apache Software Foundation and the Apache HTTP Server Project are pleased to [announce](#) the release of version 2.4.46 of the Apache HTTP Server ("httpd").

This latest release from the 2.4.x stable branch represents the best available version of Apache HTTP Server.

Apache HTTP Server version 2.4.43 or newer is required in order to operate a TLS 1.3 web server with OpenSSL 1.1.1.

[Download](#) | [ChangeLog for 2.4.46](#) | [Complete ChangeLog for 2.4](#) | [New Features in httpd 2.4](#)

### Apache httpd 2.2 End-of-Life

2018-01-01

As previously announced, the Apache HTTP Server Project has discontinued all development and patch review of the 2.2.x series of releases.

The Apache HTTP Server Project had long committed to provide maintenance releases of the 2.2.x flavor through June of 2017. The final release 2.2.34 was published in July 2017, and no further evaluation of bug reports or security risks will be considered or published for 2.2.x releases.

### Want to try out the Apache HTTP Server?

Great! We have updated our [download page](#) in an effort to better utilize our mirrors. We hope that by making it easier to use our mirrors, we will be able to provide a better download experience.

Please ensure that you [verify](#) your downloads using GPG or MD5 signatures.

### Want to contribute to the Apache HTTP Server?

Awesome! Have a look at our current 'Help Wanted' listings then:

Tasks the Apache HTTP Server Project would like help with:			
Title	Languages	Difficulty	Created
  Rewrite mod_ldap cache handling code	c	 Intermediate	2018/5/29上午1:23:20
  Learn and work on httpd's internals	c	 Advanced	2018/1/3上午1:19:40
  Websocket documentation	english,xml,lua,c	 Intermediate	2016/2/20上午1:31:35
  Improve the Request Processing guide	xml,html,c	 Advanced	2016/2/14下午7:16:11
  Traffic shaping with mod_lua	lua,c	 Advanced	2016/2/14下午5:09:50
  Make a better guide on request hooks	xml,html,c	 Advanced	2016/2/8下午5:46:07
  Write a better cache guide	xml,html	 Expert	2016/2/7上午5:30:59

Powered by [Help Wanted](#) - a task directory for Apache projects

#### Essentials

- [Download!](#)
- [About](#)
- [License](#)
- [FAQ](#)
- [Security Reports](#)

#### Source Repositories

- [General Information](#)
- [Trunk](#)
- [2.4](#)

#### Documentation

- [Version 2.4](#)
- [Trunk \(dev\)](#)
- [Wiki](#)

#### Get Involved

- [Mailing Lists](#)
- [Bug Reports](#)
- [Developer Info](#)
- [User Support](#)

#### Subprojects

- [Docs](#)
- [Test](#)
- [Flood](#)
- [libapreq](#)
- [Modules](#)
- [mod\\_fcgid](#)
- [mod\\_ftp](#)

#### Related Projects

- [Apache Traffic Server](#)
- [Apache Traffic Control](#)
- [Tomcat](#)
- [APR](#)
- [mod\\_perl](#)

#### Miscellaneous

- [Contributors](#)
- [Thanks!](#)
- [Sponsorship](#)

## Apache HTTP 服务器 2.4 文档

可用语言: [da](#) | [de](#) | [en](#) | [es](#) | [fr](#) | [ja](#) | [ko](#) | [pt-br](#) | [ru](#) | [tr](#) | [zh-cn](#) Google 搜索

## 发行说明

[Apache 2.3/2.4 的新特性](#)  
[Apache 2.1/2.2 的新特性](#)  
[Apache 2.0 的新特性](#)  
[从 2.2 升级到 2.4](#)  
[Apache 许可证](#)

## 参考手册

[编译与安装](#)  
[启动](#)  
[停止与重启](#)  
[指令快速参考](#)  
[模块](#)  
[多处理模块\(MPM\)](#)  
[过滤器](#)  
[处理器](#)  
[表达式解析器](#)  
[服务器与支持程序](#)  
[术语](#)

## 用户指南

[入门指南](#)  
[绑定指定地址与端口](#)  
[配置文件](#)  
[配置片段](#)  
[缓存指南](#)  
[内容协商](#)  
[动态共享对象\(DSO\)](#)  
[环境变量](#)  
[日志文件](#)  
[从 URL 映射到文件系统](#)  
[性能调谐](#)  
[安全技巧](#)  
[服务器全局配置](#)  
[SSL/TLS 加密](#)  
[执行 CGI 前的用户切换\(suEXEC\)](#)  
[URL 改写与 mod\\_rewrite](#)  
[虚拟主机](#)

## 索引/教程

[认证、授权与访问控制](#)  
[访问控制](#)  
[CGI 与动态内容](#)  
[.htaccess 文件](#)  
[服务器端插入\(SSII\)](#)  
[用户私人网站目录\(public\\_html\)](#)  
[反向代理设置指南](#)  
[HTTP/2 指南](#)

## 平台相关说明

[Microsoft Windows](#)  
[基于RPM安装包的系统 \(Redhat / CentOS / Fedora\)](#)  
[Novell NetWare](#)  
[EBCDIC 系统](#)

## 其它主题

[常见问题](#)  
[网站导航](#)  
[开发文档](#)  
[帮助改进文档](#)  
[其它说明](#)  
[维基](#)

可用语言: [da](#) | [de](#) | [en](#) | [es](#) | [fr](#) | [ja](#) | [ko](#) | [pt-br](#) | [ru](#) | [tr](#) | [zh-cn](#)[模块](#) | [指令](#) | [常见问题](#) | [术语](#) | [网站导航](#)

养成阅读官方文档的习惯，是最有效的学习途径。

## 2.Apache服务器

### 2.3 Apache工作模式

- Apache有prefork、worker、event三种工作模式。
  - **prefork工作模式**
    - prefork是稳定的Apache模式。
    - 在启动之初就预先派生一些子进程，然后等待客户端的请求进来，用于减少频繁创建和销毁进程的开销。
    - 每个子进程只有一个线程，在一个时间点内，只能处理一个请求。
    - 但是将请求放进队列中，一直等到有可用进程，请求才会被处理，提高了延时。
    - 通常情况下，Apache具有很强的自我调节能力，不需要额外调整。
    - 但当需要处理的并发请求较高时，需要根据业务情况进行调优。



## 2.Apache服务器

### 2.3 Apache工作模式

- Apache有prefork、worker、event三种工作模式。
  - **prefork工作模式**
    - 通常情况下，Apache具有很强的自我调节能力，不需要额外调整。但当需要处理的并发请求较高时，需要根据业务情况进行调优。
    - prefork下有StartServers、MinSpareServers、MaxSpareServers、MaxRequestWorkers四个指令用于调节父进程如何产生子进程。
      - StartServers                      初始的工作进程数
      - MinSpareServers                  空闲子进程的最小数量
      - MaxSpareServers                  空闲子进程的最大数量
      - MaxRequestWorkers                最大空闲线程数
    - 如：并发较高时，需要增加MaxRequestWorkers的值，避免更多人排队。
    - 内存较小时，需要减少MaxRequestWorkers的值，避免服务崩溃。



## 2.Apache服务器

### 2.3 Apache工作模式

- Apache有prefork、worker、event三种工作模式。
  - **worker工作模式**
    - worker模式相对于prefork来说，使用多进程和多线程混合模式。
    - Apache启动时预先分了几个子进程（数量比较少），每个子进程创建一些线程，同时包括一个监听线程。
      - 每个请求过来，会分配一个线程来进行服务。
      - 线程通常会共享父进程的内存空间，对内存占用会减少些，用线程处理会更轻量。
    - worker模式在高并发的情况下，比prefork有更多的可用进程。
      - 考虑到稳定性，worker不完全使用多线程，还引入多进程。
      - 如果使用单进程，在一个线程出错往往会导致父进程连同其它正常的子线程都出错。
      - 使用多个进程加多个线程的方式，即便某个线程出现异常，受影响的只有Apache的部分服务。





## 2.Apache服务器

### 2.3 Apache工作模式

- Apache有prefork、worker、event三种工作模式。
  - event工作模式
    - event和worker模式较为相似，但event解决了keep-alive场景下线程长期被占用而造成的资源浪费问题。
    - event模式中，会有一个专门的线程来管理keep-alive类型的线程。
      - 当有真实请求时将请求传递给服务线程，执行完毕后释放。
      - 增强了高并发场景下的请求处理能力。

根据业务特性选择Apache的工作模式，进行Web服务器调优。



## 2.Apache服务器

### 2.4 Apache Module

- Apache是模块化的设计，大多数功能被分散到各模块中，各模块在系统启动时按需载入。
  - 安装Apache时会默认安装一些模块，如果需要实现某种特定的功能可以根据实际需求自行安装Apache模块。
  - Apache模块只与Apache版本有关，和操作系统无关。

Apache Module

内置模块：随Apache而安装，自行启用/禁用

扩展模块：根据需要而自行安装，自由管理

自定义模块：根据自身业务需要定制开发的模块



```
[root@Project-03-Task-01 ~]# yum repolist
```

```
仓库标识                               仓库名称
appstream                               CentOS Linux 8 - AppStream
baseos                                  CentOS Linux 8 - BaseOS
extras                                  CentOS Linux 8 - Extras
```

```
[root@Project-03-Task-01 ~]#
```

```
[root@Project-03-Task-01 ~]# yum install epel-release
```

```
上次元数据过期检查: 0:04:20 前, 执行于 2021年03月18日 星期四 19时42分21秒。
```

```
依赖关系解决。
```

```
=====
软件包                                架构                版本                仓库                大小
=====
安装:
epel-release                           noarch               8-8.el8             extras                23 k
```

```
事务概要
```

```
安装 1 软件包
```

```
总下载: 23 k
```

```
安装大小: 32 k
```

```
确定吗? [y/N]: y
```

```
下载软件包:
```

```
epel-release-8-8.el8.noarch.rpm                                               103 kB/s | 23 kB    00:00
```

```
总计
```

```
24 kB/s | 23 kB    00:00
```

```
运行事务检查
```

```
事务检查成功。
```

```
运行事务测试
```

```
事务测试成功。
```

```
运行事务
```

```
准备中 :
```

```
安装      : epel-release-8-8.el8.noarch
```

```
运行脚本 : epel-release-8-8.el8.noarch
```

```
验证      : epel-release-8-8.el8.noarch
```

```
1/1
```

```
1/1
```

```
1/1
```

```
1/1
```

```
已安装:
```

```
epel-release-8-8.el8.noarch
```

```
完毕!
```

```
[root@Project-03-Task-01 ~]# yum repolist
```

```
仓库标识                               仓库名称
appstream                               CentOS Linux 8 - AppStream
baseos                                  CentOS Linux 8 - BaseOS
epel                                     Extra Packages for Enterprise Linux 8 - x86_64
epel-modular                            Extra Packages for Enterprise Linux Modular 8 - x86_64
extras                                  CentOS Linux 8 - Extras
```

```
[root@Project-03-Task-01 ~]#
```

```
[root@Project-03-Task-01 ~]#
```

```
[root@Project-03-Task-01 ~]# yum search mod_
```

```
上次元数据过期检查: 0:01:40 前, 执行于 2021年03月18日 星期四 19时53分26秒。
```

```
===== 名称 和 概况 匹配: mod_ =====
certwatch-mod_ssl.x86_64 : SSL/TLS certificate expiry warnings for mod_ssl
mod_auth_mellon-diagnostics.x86_64 : Build of mod_auth_mellon with diagnostic logging
mod_perl-devel.x86_64 : Files needed for building XS modules that use mod_perl
===== 名称 匹配: mod_ =====
ipv6calc-mod_ipv6calc.x86_64 : Apache module for ipv6calc
lighttpd-mod_authn_gssapi.x86_64 : Authentication module for lighttpd that uses GSSAPI
lighttpd-mod_authn_mysql.x86_64 : Authentication module for lighttpd that uses a MySQL database
lighttpd-mod_authn_pam.x86_64 : Authentication module for lighttpd that uses PAM
lighttpd-mod_mysql_vhost.x86_64 : Virtual host module for lighttpd that uses a MySQL database
mod_auth_gssapi.x86_64 : A GSSAPI Authentication module for Apache
mod_auth_mellon.x86_64 : A SAML 2.0 authentication module for the Apache Httpd Server
mod_authnz_pam.x86_64 : PAM authorization checker and PAM Basic Authentication provider
mod_dav_svn.x86_64 : Apache httpd module for Subversion server
mod_fcgid.x86_64 : FastCGI interface module for Apache 2
mod_flv.x86_64 : FLV progressive download streaming for the Apache HTTP Server
mod_http2.x86_64 : module implementing HTTP/2 for Apache 2
mod_intercept_form_submit.x86_64 : Apache module to intercept login form submission and run PAM authentication
mod_ldap.x86_64 : LDAP authentication modules for the Apache HTTP Server
mod_limitipconn.x86_64 : Simultaneous connection limiting module for Apache
mod_lookup_identity.x86_64 : Apache module to retrieve additional information about the authenticated user
mod_md.x86_64 : Certificate provisioning using ACME for the Apache HTTP Server
mod_passenger.x86_64 : Apache Module for Phusion Passenger
mod_perl.x86_64 : An embedded Perl interpreter for the Apache HTTP Server
mod_proxy_html.x86_64 : HTML and XML content filters for the Apache HTTP Server
mod_qos.x86_64 : Quality of service module for Apache
mod_security.x86_64 : Security module for the Apache HTTP Server
mod_security-mlogc.x86_64 : ModSecurity Audit Log Collector
mod_security-crs.noarch : ModSecurity Rules
mod_session.x86_64 : Session interface for the Apache HTTP Server
mod_speedycgi.x86_64 : SpeedyCGI module for the Apache HTTP Server
mod_ssl.x86_64 : SSL/TLS module for the Apache HTTP Server
python3-mod_wsgi.x86_64 : A WSGI interface for Python web applications in Apache
python38-mod_wsgi.x86_64 : A WSGI interface for Python web applications in Apache
===== 概况 匹配: mod_ =====
gridsite-libs.x86_64 : Run time libraries for mod_gridsite and gridsite-clients
perl-Apache-DBI.noarch : Persistent database connections with Apache/mod_perl
perl-Apache2-SOAP.noarch : A replacement for Apache::SOAP designed to work with mod_perl 2
perl-JSON-RPC-Apache2.noarch : JSON-RPC server for mod_perl2
```

```
[root@Project-03-Task-01 ~]#
```

```
[root@Project-03-Task-01 ~]# yum install mod_http2
```

```
上次元数据过期检查: 0:02:22 前, 执行于 2021年03月18日 星期四 19时53分26秒。
```

```
依赖关系解决。
```

```
===== 软件包 架构 版本 仓库 大小 =====
安装:
mod_http2                               x86_64                               1.15.7-2.module_el8.3.0+477+498bb568  appstream 154 k
安装依赖关系:
apr                                       x86_64                               1.6.3-11.el8                          appstream 125 k
apr-util                                  x86_64                               1.6.1-6.el8                             appstream 105 k
centos-logos-httpd                       noarch                                80.5-2.el8                              baseos   24 k
httpd                                     x86_64                               2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream 1.7 M
httpd-filesystem                         noarch                                2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream 37 k
httpd-tools                              x86_64                               2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream 104 k
mailcap                                  noarch                                2.1.48-3.el8                             baseos   39 k
安装弱依赖:
apr-util-bdb                             x86_64                               1.6.1-6.el8                             appstream 25 k
apr-util-openssl                         x86_64                               1.6.1-6.el8                             appstream 27 k
启用模块流:
httpd                                     2.4
```

## 2.Apache服务器

### 2.4 Apache Module

- Apache模块的管理
  - 安装成功后的Apache模块存放目录：
    - CentOS操作系统：
      - 存放目录为：/usr/lib64/httpd/modules
      - 模块文件名：模块名.so
    - Ubuntu Server操作系统：
      - 存放目录为：/etc/apache2/mods-available
      - 已启用模块存放目录为：/etc/apache2/mods-enabled
      - 模块文件名：模块名.load （每一个模块对应一个.conf文件）

不同Linux发行版，安装应用软件后的配置文件、管理方式都有不同。

查阅资料和技术交流讨论时，请务必明确：

操作系统发行版类型+操作系统版本号+应用软件名+应用软件版本号



```
[root@Project-03-Task-01 ~]# yum install httpd
```

```
上次元数据过期检查: 0:04:27 前, 执行于 2021年03月18日 星期四 19时53分26秒。
```

```
依赖关系解决。
```

```
=====
软件包                                架构                                版本                                仓库                                大小
=====
安装:
httpd                                  x86_64                              2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream                            1.7 M
安装依赖关系:
apr                                     x86_64                              1.6.3-11.el8                          appstream                            125 k
apr-util                               x86_64                              1.6.1-6.el8                            appstream                            105 k
centos-logos-httpd                    noarch                               80.5-2.el8                             baseos                                24 k
httpd-filesystem                      noarch                               2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream                            37 k
httpd-tools                            x86_64                              2.4.37-30.module_el8.3.0+561+97fdbbcc  appstream                            104 k
mailcap                                noarch                               2.1.48-3.el8                           baseos                                39 k
mod_http2                              x86_64                              1.15.7-2.module_el8.3.0+477+498bb568  appstream                            154 k
安装弱的依赖:
apr-util-bdb                           x86_64                              1.6.1-6.el8                            appstream                            25 k
apr-util-openssl                       x86_64                              1.6.1-6.el8                            appstream                            27 k
启用模块流:
httpd                                  2.4
=====
```

```
事务概要
```

```
=====
安装 10 软件包
```

```
总下载: 2.3 M
```

```
安装大小: 6.0 M
```

```
确定吗? [y/N]: y
```

```
下载软件包:
```

```
(1/10): apr-util-bdb-1.6.1-6.el8.x86_64.rpm                                357 kB/s | 25 kB  00:00
(2/10): apr-util-openssl-1.6.1-6.el8.x86_64.rpm                          1.2 MB/s | 27 kB  00:00
(3/10): apr-util-1.6.1-6.el8.x86_64.rpm                                  1.1 MB/s | 105 kB  00:00
(4/10): apr-1.6.3-11.el8.x86_64.rpm                                       1.2 MB/s | 125 kB  00:00
(5/10): httpd-filesystem-2.4.37-30.module_el8.3.0+561+97fdbbcc.noarch.rpm  2.5 MB/s | 37 kB  00:00
(6/10): httpd-tools-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm      4.0 MB/s | 104 kB  00:00
(7/10): mod_http2-1.15.7-2.module_el8.3.0+477+498bb568.x86_64.rpm        3.2 MB/s | 154 kB  00:00
(8/10): httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64.rpm           15 MB/s | 1.7 MB  00:00
(9/10): centos-logos-httpd-80.5-2.el8.noarch.rpm                          153 kB/s | 24 kB  00:00
(10/10): mailcap-2.1.48-3.el8.noarch.rpm                                    182 kB/s | 39 kB  00:00
```

```
总计
```

```
1.5 MB/s | 2.3 MB  00:01
```

```
运行事务检查
```

```
事务检查成功。
```

```
运行事务测试
```

```
事务测试成功。
```

```
运行事务
```

```
准备中 :
```

```
安装      : apr-1.6.3-11.el8.x86_64                                          1/1
运行脚本 : apr-1.6.3-11.el8.x86_64                                          1/10
安装      : apr-util-bdb-1.6.1-6.el8.x86_64                                2/10
安装      : apr-util-openssl-1.6.1-6.el8.x86_64                            3/10
安装      : apr-util-1.6.1-6.el8.x86_64                                    4/10
运行脚本 : apr-util-1.6.1-6.el8.x86_64                                    4/10
安装      : httpd-tools-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64       5/10
安装      : mailcap-2.1.48-3.el8.noarch                                       6/10
安装      : centos-logos-httpd-80.5-2.el8.noarch                            7/10
运行脚本 : httpd-filesystem-2.4.37-30.module_el8.3.0+561+97fdbbcc.noarch   8/10
安装      : httpd-filesystem-2.4.37-30.module_el8.3.0+561+97fdbbcc.noarch  8/10
安装      : mod_http2-1.15.7-2.module_el8.3.0+477+498bb568.x86_64          9/10
安装      : httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64            10/10
运行脚本 : httpd-2.4.37-30.module_el8.3.0+561+97fdbbcc.x86_64            10/10
验证      : apr-1.6.3-11.el8.x86_64                                          1/10
验证      : apr-util-1.6.1-6.el8.x86_64                                    2/10
```

```
[root@Project-03-Task-01 ~]# cd /usr/lib64/httpd/modules/
```

```
[root@Project-03-Task-01 modules]#
```

```
[root@Project-03-Task-01 modules]# ls
```

```
mod_access_compat.so  mod_authn_file.so      mod_buffer.so          mod_dbd.so             mod_heartbeat.so       mod_log_forensic.so    mod_proxy_balancer.so  mod_proxy_uwsgi.so     mod_slotmem_shm.so     mod_userdir.so
mod_actions.so        mod_authn_socache.so   mod_cache_disk.so     mod_deflate.so        mod_heartmonitor.so   mod_logio.so           mod_proxy_connect.so   mod_proxy_wstunnel.so  mod_socache_dbm.so     mod_usertrack.so
mod_alias.so          mod_authz_core.so      mod_cache.so          mod_dialup.so         mod_http2.so          mod_lua.so             mod_proxy_express.so   mod_ratelimit.so      mod_socache_memcache.so  mod_version.so
mod_allowmethods.so  mod_authz_dbd.so       mod_cache_socache.so  mod_dir.so            mod_include.so        mod_macro.so           mod_proxy_fcgi.so      mod_reflector.so      mod_socache_shmcb.so    mod_vhost_alias.so
mod_asis.so          mod_authz_dbm.so       mod_cgid.so           mod_dumpio.so         mod_info.so           mod_mime_magic.so      mod_proxy_fdpass.so    mod_remotepip.so      mod_spelling.so         mod_watchdog.so
mod_auth_basic.so    mod_authz_groupfile.so mod_cgid.so            mod_echo.so           mod_lbmethod_bybusyness.so  mod_mime.so           mod_proxy_ftp.so       mod_reqtimeout.so     mod_status.so
mod_auth_digest.so  mod_authz_host.so      mod_charset_lite.so   mod_env.so            mod_lbmethod_byrequests.so  mod_mpm_event.so      mod_proxy_hcheck.so    mod_request.so        mod_substitute.so
mod_authn_anon.so   mod_authz_owner.so     mod_data.so           mod_expires.so        mod_lbmethod_bytraffic.so  mod_mpm_prefork.so    mod_proxy_http2.so     mod_rewrite.so        mod_suexec.so
mod_authn_core.so   mod_authz_user.so      mod_dav_fs.so         mod_ext_filter.so     mod_lbmethod_heartbeat.so  mod_mpm_worker.so     mod_proxy_http.so      mod_sed.so            mod_systemd.so
mod_authn_dbd.so    mod_autoindex.so       mod_dav_lock.so       mod_filter.so         mod_log_config.so      mod_negotiation.so    mod_proxy_scgi.so      mod_setenvif.so      mod_unique_id.so
mod_authn_dbm.so    mod_brotli.so          mod_dav.so            mod_headers.so        mod_log_debug.so       mod_proxy_ajp.so       mod_proxy.so           mod_slotmem_plain.so  mod_unixd.so
```

```
[root@Project-03-Task-01 modules]#
```

```
[root@Project-03-Task-01 modules]# ls -l
```

```
总用量 3088
```

```
--wxr-xr-x. 1 root root 11984 11月 4 11:21 mod_access_compat.so
--wxr-xr-x. 1 root root 11920 11月 4 11:21 mod_actions.so
--wxr-xr-x. 1 root root 20240 11月 4 11:21 mod_alias.so
--wxr-xr-x. 1 root root 11896 11月 4 11:21 mod_allowmethods.so
--wxr-xr-x. 1 root root 11848 11月 4 11:21 mod_asis.so
--wxr-xr-x. 1 root root 16120 11月 4 11:21 mod_auth_basic.so
--wxr-xr-x. 1 root root 36824 11月 4 11:21 mod_auth_digest.so
--wxr-xr-x. 1 root root 11912 11月 4 11:21 mod_authn_anon.so
--wxr-xr-x. 1 root root 16128 11月 4 11:21 mod_authn_core.so
--wxr-xr-x. 1 root root 16032 11月 4 11:21 mod_authn_dbd.so
--wxr-xr-x. 1 root root 11944 11月 4 11:21 mod_authn_dbm.so
--wxr-xr-x. 1 root root 11928 11月 4 11:21 mod_authn_file.so
--wxr-xr-x. 1 root root 20304 11月 4 11:21 mod_authn_socache.so
--wxr-xr-x. 1 root root 24496 11月 4 11:21 mod_authz_core.so
--wxr-xr-x. 1 root root 16072 11月 4 11:21 mod_authz_dbd.so
--wxr-xr-x. 1 root root 16072 11月 4 11:21 mod_authz_dbm.so
--wxr-xr-x. 1 root root 16056 11月 4 11:21 mod_authz_groupfile.so
--wxr-xr-x. 1 root root 16064 11月 4 11:21 mod_authz_host.so
--wxr-xr-x. 1 root root 11888 11月 4 11:21 mod_authz_owner.so
--wxr-xr-x. 1 root root 11912 11月 4 11:21 mod_authz_user.so
--wxr-xr-x. 1 root root 40832 11月 4 11:21 mod_autoindex.so
--wxr-xr-x. 1 root root 20200 11月 4 11:21 mod_brotli.so
--wxr-xr-x. 1 root root 11920 11月 4 11:21 mod_buffer.so
--wxr-xr-x. 1 root root 36848 11月 4 11:21 mod_cache_disk.so
--wxr-xr-x. 1 root root 78664 11月 4 11:21 mod_cache.so
--wxr-xr-x. 1 root root 36832 11月 4 11:21 mod_cache_socache.so
--wxr-xr-x. 1 root root 49184 11月 4 11:21 mod_cgid.so
--wxr-xr-x. 1 root root 32600 11月 4 11:21 mod_cgi.so
--wxr-xr-x. 1 root root 24344 11月 4 11:21 mod_charset_lite.so
--wxr-xr-x. 1 root root 11848 11月 4 11:21 mod_data.so
--wxr-xr-x. 1 root root 58304 11月 4 11:21 mod_dav_fs.so
--wxr-xr-x. 1 root root 28800 11月 4 11:21 mod_dav_lock.so
--wxr-xr-x. 1 root root 107560 11月 4 11:21 mod_dav.so
--wxr-xr-x. 1 root root 24376 11月 4 11:21 mod_dbd.so
--wxr-xr-x. 1 root root 36720 11月 4 11:21 mod_deflate.so
--wxr-xr-x. 1 root root 16032 11月 4 11:21 mod_dialup.so
--wxr-xr-x. 1 root root 16032 11月 4 11:21 mod_dir.so
--wxr-xr-x. 1 root root 11944 11月 4 11:21 mod_dumpio.so
--wxr-xr-x. 1 root root 11920 11月 4 11:21 mod_echo.so
--wxr-xr-x. 1 root root 11936 11月 4 11:21 mod_env.so
--wxr-xr-x. 1 root root 16056 11月 4 11:21 mod_expires.so
--wxr-xr-x. 1 root root 28400 11月 4 11:21 mod_ext_filter.so
--wxr-xr-x. 1 root root 20176 11月 4 11:21 mod_filter.so
--wxr-xr-x. 1 root root 24512 11月 4 11:21 mod_headers.so
--wxr-xr-x. 1 root root 11936 11月 4 11:21 mod_heartbeat.so
--wxr-xr-x. 1 root root 28432 11月 4 11:21 mod_heartmonitor.so
--wxr-xr-x. 1 root root 260184 9月 1 2020 mod_http2.so
```

Apache Httpd Server 安装的模块

```
[root@Project-03-Task-01 ~]# ls /etc/httpd/conf.modules.d/
00-base.conf 00-dav.conf 00-lua.conf 00-mpm.conf 00-optional.conf 00-proxy.conf 00-systemd.conf 01-cgi.conf 10-h2.conf 10-proxy_h2.conf README
[root@Project-03-Task-01 ~]#
[root@Project-03-Task-01 ~]# cat /etc/httpd/conf.modules.d/00-systemd.conf
# This file configures systemd module:
LoadModule systemd_module modules/mod_systemd.so
[root@Project-03-Task-01 ~]#
[root@Project-03-Task-01 ~]# cat /etc/httpd/conf.modules.d/10-h2.conf
LoadModule http2_module modules/mod_http2.so
[root@Project-03-Task-01 ~]#
[root@Project-03-Task-01 ~]# cat /etc/httpd/conf.modules.d/00-proxy.conf
# This file configures all the proxy modules:
LoadModule proxy_module modules/mod_proxy.so
LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_express_module modules/mod_proxy_express.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
LoadModule proxy_fdpass_module modules/mod_proxy_fdpass.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_hcheck_module modules/mod_proxy_hcheck.so
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
LoadModule proxy_uwsgi_module modules/mod_proxy_uwsgi.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
[root@Project-03-Task-01 ~]#
[root@Project-03-Task-01 ~]#
```

## Apache Httpd Server模块的使用



Apache 2.4.37 常用模块列表

序号	模块名	功能说明	默认安装
1	mod_actions	运行基于 MIME 类型的 CGI 脚本	是
2	mod_alias	提供从文件系统的不同部分到文档树的映射和 URL 重定向	是
3	mod_asis	原样发送文档信息，而不添加常用的 HTTP 头	是
4	mod_auth_basic	使用基本认证	是
5	mod_auth_digest	使用 MD5 加密算法进行验证	否
7	mod_authn_anon	允许匿名用户访问认证的区域	否
8	mod_authn_dbd	使用数据库保存用户验证信息	否
9	mod_authn_dbm	使用 DBM 数据文件保存用户验证信息	否
10	mod_authn_default	在未正确配置认证模块的情况下拒绝一切认证	是
11	mod_authz_groupfile	使用 plaintext 文件进行组验证	是
12	mod_authz_file	使用文本文件保存用户验证信息	是
13	mod_authnz_ldap	允许使用 LDAP 目录存储用户名和密码执行 HTTP 基本身份验证	否
14	mod_authz_host	提供基于主机名称或 IP 地址的访问限制	是
15	mod_authz_user	提供基于用户的访问限制	是
16	mod_autoindex	自动生成目录索引，类似于 Unix 的 ls、Windows 的 dir 命令	是
17	mod_cache	兼容 RFC 2616 标准的 HTTP 缓存过滤器	否
18	mod_cgi	在非线程型 MPM(prefork)上提供对 CGI 脚本执行的支持	是
19	mod_cgid	在线程型 MPM(worker)上用一个外部 CGI 守护进程执行 CGI 脚本	是
20	mod_dir	指定目录索引文件以及为目录提供“尾斜杠”重定向	是



Apache 2.4.37 常用模块列表

序号	模块名	功能说明	默认安装
21	mod_env	允许 Apache 修改或清除传送到 CGI 脚本和 SSI 页面的环境变量	是
22	mod_example_hooks	提供编写 Apache API 模块的示例	否
23	mod_filter	根据上下文实际情况对过滤器动态配置	是
24	mod_imagemap	处理服务器端图像映射	是
25	mod_include	实现服务端包含文档 (SSI) 的解析	是
26	mod_isapi	仅限于在 Windows 平台上实现 ISAPI 扩展	是
27	mod_ldap	使用第三方 LDAP 模块进行 LDAP 链接服务	否
28	mod_log_config	允许记录日志和定制日志文件格式	是
29	mod_logio	记录每个请求的输入、输出的字节数	否
30	mod_mime	根据文件扩展名决定应答的行为和内容	是
31	mod_negotiation	提供内容选择 (content negotiation, 从几个有效文档中选择一个最匹配客户端要求的文档的过程)	是
32	mod_nw_ssl	支持在 NetWare 平台上实现 SSL 加密	是
33	mod_proxy	支持 HTTP1.1 协议的代理和网关服务器	否
34	mod_proxy_ajp	mod_proxy 的 AJP 支持模块	否
35	mod_proxy_balancer	mod_proxy 的负载均衡模块	否



Apache 2.4.37 常用模块列表

序号	模块名	功能说明	默认安装
36	mod_proxy_ftp	mod_proxy 的 FTP 支持模块	否
37	mod_proxy_http	mod_proxy 的 HTTP 支持模块	否
38	mod_setenvif	允许设置基于请求的环境变量	是
39	mod_so	允许运行时加载 DSO 模块	否
40	mod_ssl	使用 SSL 和 TLS 的加密	否
41	mod_status	提供服务器性能运行信息	是
42	mod_userdir	设置每个用户的网站目录	是
43	mod_usertrack	记录用户在网站上的活动	否
44	mod_vhost_alias	提供大量虚拟机的动态配置	否
45	mod_proxy_fcgi	提供对 fcgi 的代理	否
46	mod_ratelimit	限制用户带宽	否
47	mod_request	请求模块，对请求做过滤	是
48	mod_remoteip	用来匹配客户端的 IP 地址	是



## 3. 使用Apache发布网站

### 3.1 任务1

任务1：安装Apache与服务管理

任务2：使用Apache发布静态网站

任务3：使用Apache发布多个静态网站



## 3. 使用Apache发布网站

### 3.1 任务1

#### 任务1：安装Apache与服务管理

步骤1：创建虚拟机并完成CentOS的安装

步骤2：完成虚拟机的主机配置、网络配置及通信测试

步骤3：通过在线方式安装Apache

步骤4：启动Apache服务

步骤5：查看Apache运行信息

步骤6：配置httpd服务为开机自启动





操作视频 / 现场演示

## ✓ 任务1: 安装Apache与服务管理

### ■ 任务目标:

- Apache HTTP Server的服务httpd已启动
- Apache HTTP Server的服务httpd设置为开机自启动





## 命令指南 / 操作引导

1. #使用yum工具安装Apache
2. root@Project-03-Task-01 ~]# yum install -y httpd
3. #使用systemctl start命令启动httpd服务
4. [root@Project-03-Task-01 ~]# systemctl start httpd
5. #使用systemctl status命令查看httpd服务
6. [root@Project-03-Task-01 ~]# systemctl status httpd
7. #命令systemctl enable可设置某服务为开机自启动。
8. #命令systemctl disable可设置某服务为开机不自动启动。
9. [root@Project-03-Task-01 ~]# systemctl enable httpd.service
10. Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service /usr/lib/systemd/system/httpd.service.
11. #使用systemctl list-unit-files命令确认httpd服务是否已配置为开机自启动
12. [root@Project-03-Task-01 ~]# systemctl list-unit-files | grep httpd.service
13. #下述信息说明httpd.service已配置为开机自启动
14. httpd.service                    enabled



## 3. 使用Apache发布网站

### 3.2 任务2

#### 任务2：使用Apache发布静态网站

步骤1：创建网站目录与网站内容

步骤2：发布静态网站 Site1

步骤3：使用本地主机浏览器访问网站 Site1







操作视频 / 现场演示



- ✓ 任务2：使用Apache发布静态网站
  - 任务目标：
    - 本地主机浏览器正常浏览 Site1 的内容





## 命令指南 / 操作引导

1. #使用echo指令快速创建网站Site1的网站首页
2. [root@Project-03-Task-01 ~]# echo "<h1>Site1。 http://10.10.2.104</h1>" > /var/www/html/index.html
3. #配置文件/etc/httpd/conf/httpd.conf的内容
4. [root@Project-03-Task-01 ~]# cat /etc/httpd/conf/httpd.conf
5. #httpd.conf配置文件内容较多，本部分仅显示与默认网站配置有关的内容
6. #默认网站配置
7. Listen 80
8. #定义默认网站路径
9. DocumentRoot "/var/www/html"
10. <Directory "/var/www/html">
11. #网站目录默认开启Indexes、FollowSymLinks服务器特性。
12. #目录下无index文件，则允许显示该目录下的文件，并跟踪符号链接
13. Options Indexes FollowSymLinks
14. #其它配置文件中出现对80端口的配置且与本处配置相冲突，以此处为准
15. AllowOverride None
16. #允许所有地址访问
17. Require all granted
18. </Directory>
19. #为使网站能正常访问，本任务暂时关闭防火墙等安全措施。
20. #使用systemctl stop命令关闭防火墙
21. [root@Project-03-Task-01 ~]# systemctl stop firewalld
22. #使用setenforce命令将SELinux设置为permissive模式
23. [root@Project-03-Task-01 ~]# setenforce 0



## 3. 使用Apache发布网站

### 3.3 任务3

#### 任务3：使用Apache发布多个静态网站

静态网站规划表

网站名	访问地址	网站存放目录
Site1	http://10.10.2.104	/var/www/html
Site2	http://10.10.2.104/aliasA	/var/www/html/site2
Site3	http://10.10.2.104/aliasB	/var/www/html/site3
Site4	http://10.10.2.104:81	/var/www/html/site4
Site5	http://10.10.2.104:82	/var/www/html/site5
Site6	http://www.domain1.com	/var/www/html/site6
Site7	http://www.domain2.com	/var/www/html/site7



## 3. 使用Apache发布网站

### 3.3 任务3

#### 任务3：使用Apache发布多个静态网站

步骤1：创建网站目录与网站内容

步骤2：发布网站 Site1

步骤3：发布网站 Site2、Site3

步骤4：发布网站 Site4、Site5

步骤5：发布网站 Site6、Site7

步骤6：使用本地主机浏览器访问网站 Site1 ~ Site7





操作视频 / 现场演示



- ✓ 任务3：使用Apache发布多个静态网站
  - 任务目标：
    - 本地主机浏览器正常浏览 Site1~Site7 的内容





## 命令指南 / 操作引导

1. #使用echo指令快速创建网站Site1的网站首页  
2. [root@Project-03-Task-01 ~]# echo "<h1>Site1。 http://10.10.2.104</h1>" > /var/www/html/index.html
3. #创建网站Site2的目录和网站首页  
4. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site2  
5. [root@Project-03-Task-01 ~]# echo "<h1>Site2。 http://10.10.2.104/aliasA</h1>" > /var/www/html/site2/index.html
6. #创建网站Site3的目录和网站首页  
7. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site3  
8. [root@Project-03-Task-01 ~]# echo "<h1>Site3。 http://10.10.2.104/aliasB</h1>" > /var/www/html/site3/index.html
9. #创建网站Site4的目录和网站首页  
10. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site4  
11. [root@Project-03-Task-01 ~]# echo "<h1>Site4。 http://10.10.2.104:81</h1>" > /var/www/html/site4/index.html
12. #创建网站Site5的目录和网站首页  
13. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site5  
14. [root@Project-03-Task-01 ~]# echo "<h1>Site5。 http://10.10.2.104:82</h1>" > /var/www/html/site5/index.html
15. #创建网站Site6的目录和网站首页  
16. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site6  
17. [root@Project-03-Task-01 ~]# echo "<h1>Site6。 http://www.domain1.com</h1>" > /var/www/html/site6/index.html
18. #创建网站Site7的目录和网站首页  
19. [root@Project-03-Task-01 ~]# mkdir /var/www/html/site7  
20. [root@Project-03-Task-01 ~]# echo "<h1>Site7。 http://www.domain2.com</h1>" > /var/www/html/site7/index.html





## 命令指南 / 操作引导

1. #网站Site2、Site3通过在Apache默认网站上增加虚拟目录aliasA、aliasB来发布
2. [root@Project-03-Task-01 ~]# vi /etc/httpd/conf/httpd.conf
3. #httpd.conf配置文件内容较多，本部分仅显示与网站配置有关的内容
4. #新增Site2的配置信息，通过aliasA发布网站
5. Alias /aliasA "/var/www/html/site2"
6. #定义aliasA对应的网站路径
7. <Directory "/var/www/html/site2">
8. AllowOverride None
9. #目录不启用任何服务器特性
10. Options None
11. #允许所有地址访问
12. Require all granted
13. </Directory>
14. #新增Site3的配置信息，通过aliasB发布网站
15. Alias /aliasB "/var/www/html/site3"
16. #定义aliasB对应的网站路径
17. <Directory "/var/www/html/site3">
18. AllowOverride None
19. #目录不启用任何服务器特性
20. Options None
21. #允许所有地址访问
22. Require all granted
23. </Directory>
24. #使用systemctl reload命令重新载入Apache配置文件
25. [root@Project-03-Task-01 ~]# systemctl reload httpd





## 命令指南 / 操作引导

1. #网站Site4、Site5通过不同的端口发布
2. #在Apache额外配置目录中创建新的配置文件，发布网站Site4、Site5。  
[root@Project-03-Task-01 ~]#vi /etc/httpd/conf.d/siteport.conf
4. #新增Site4的配置信息，通过81端口发布网站
5. Listen 81
6. #定义81端口对应的网站路径
7. <VirtualHost \*:81>
8.     DocumentRoot /var/www/html/site4
9. </VirtualHost>
10. #新增Site5的配置信息，通过82端口发布网站
11. Listen 82
12. #定义82端口对应的网站路径
13. <VirtualHost \*:82>
14.     DocumentRoot /var/www/html/site5
15. </VirtualHost>
16. #使用systemctl reload命令重新载入Apache配置文件
17. [root@Project-03-Task-01 ~]# systemctl reload httpd







## 命令指南 / 操作引导

1. #网站Site6、Site7通过不同的域名发布，但使用标准80端口
2. #在Apache额外配置目录中创建新的配置文件，发布网站Site6、Site7。  
[root@Project-03-Task-01 ~]#vi /etc/httpd/conf.d/sitedomain.conf
4. #新增Site6的配置信息，通过域名www.domain1.com发布网站
5. <VirtualHost \*:80>
6. #域名设置为www.domain1.com
7. ServerName www.domain1.com
8. #绑定domain1.com域名
9. ServerAlias domain1.com
10. #定义对应的网站路径
11. DocumentRoot /var/www/html/site6
12. </VirtualHost>
13. #新增Site7的配置信息，通过域名www.domain2.com发布网站
14. <VirtualHost \*:80>
15. #域名设置为www.domain2.com
16. ServerName www.domain2.com
17. #绑定domain2.com域名
18. ServerAlias domain2.com
19. #定义对应的网站路径
20. DocumentRoot /var/www/html/site7
21. </VirtualHost>
22. #使用systemctl reload命令重新载入Apache配置文件
23. [root@Project-03-Task-01 ~]# systemctl reload httpd





任务总结 / 任务扩展

- ✓ 任务3：使用Apache发布多个静态网站
  - Site1：使用Apache默认网站，网页放默认目录即可
  - Site2 - Site3：
    - 使用虚拟目录发布
    - 默认网站配置中增加alias配置
  - Site4 - Site5：
    - 使用同一IP地址，选择非标准端口发布
    - 创建网站配置文件，每个网站采用独有端口
  - Site6 - Site7：
    - 使用同一IP地址和相同端口，不同域名发布
    - 创建网站配置文件，每个网站定义独有域名



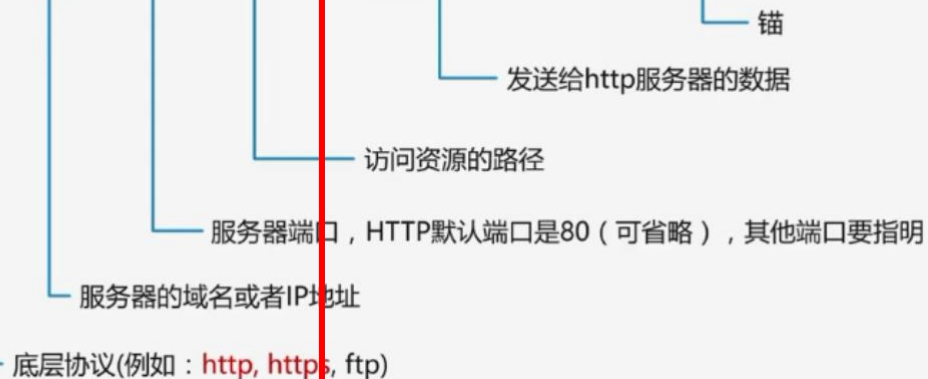


任务总结 / 任务扩展

### ✓ 任务3：使用Apache发布多个静态网站

URL的详细格式

`schema://host[:port#]/path/./[?query-string][#anchor]`



只要有一个字段信息独有  
该网站就能够被区分出来



## 4. LAMP

- LAMP是发布PHP程序的开源稳定架构，由Linux作为操作系统、Apache作为网站服务器、MySQL/MariaDB作为数据库管理系统、PHP/Perl/Python作为服务器端脚本解释器。
- 本项目使用CentOS操作系统，使用Apache网站服务器，使用MariaDB数据库管理系统，使用PHP服务器端脚本解释器，实现LAMP架构的网站服务器部署。



## 4. LAMP

---

### 4.1 任务4

任务4: 实现LAMP

任务5: 通过WordPress建设内容网站

任务6: 查看Apache的运行状态



# 4. LAMP

## 4.1 任务4

### 任务4：实现LAMP

步骤1：创建虚拟机并完成CentOS的安装

步骤2：完成虚拟机的主机配置、网络配置及通信测试、关闭安全措施

步骤3：完成Apache的安装配置

步骤4：完成MariaDB的安装配置

步骤5：完成PHP的安装配置

步骤6：LAMP部署测试





操作视频 / 现场演示

## ✓ 任务4：实现LAMP

### ■ 任务目标：

- 实现LAMP部署
- 在本地主机上正常浏览测试页面





## 命令指南 / 操作引导

1. #使用yum工具安装MariaDB
2. [root@Project-03-Task-03 ~]# yum install -y mariadb-server
3. #启动mariadb服务，并设置为开机自启动
4. [root@Project-03-Task-03 ~]# systemctl start mariadb
5. [root@Project-03-Task-03 ~]# systemctl enable mariadb
6. #设置MariaDB数据库管理系统root账户的密码为centos@mariadb#123
7. [root@Project-03-Task-03 ~]# mysqladmin -uroot password 'centos@mariadb#123'
8. #使用yum module list php命令列出库中所有PHP模块
9. [root@Project-03-Task-03 ~]# yum module list php
10. #使用yum module enable命令启用库中的PHP 7.4软件
11. [root@Project-03-Task-03 ~]# yum module -y enable php:7.4
12. #使用yum工具安装PHP 7.4解析器
13. [root@Project-03-Task-03 ~]# yum install -y php
14. #使用yum工具安装php-mysqldb模块
15. [root@Project-03-Task-03 ~]# yum install -y php-mysqldb







## 命令指南 / 操作引导

1. #验证Apache的httpd服务是否正常启动
2. [root@Project-03-Task-03 ~]# systemctl status httpd
3. #验证MariaDB的mariadb服务是否正常启动
4. [root@Project-03-Task-03 ~]# systemctl status mariadb
5. #验证PHP解析器是否安装
6. [root@Project-03-Task-03 ~]# php -v
7. #验证Apache的httpd服务是否设置为开机自启动
8. [root@Project-03-Task-03 ~]# systemctl list-unit-files | grep httpd.service
9. #验证MariaDB的mariadb服务是否设置为开机自启动
10. [root@Project-03-Task-03 ~]# systemctl list-unit-files | grep mariadb.service
11. #使用echo指令在Apache默认网站下快速创建PHP测试程序
12. [root@Project-03-Task-03 ~]# echo "<?php phpinfo(); ?>" > /var/www/html/test.php



## 4. LAMP

### 4.2 任务5

#### 任务5：通过WordPress建设内容网站

步骤1：验证是否满足WordPress部署要求

步骤2：部署前准备工作：创建数据库 安装php-json模块

步骤3：获取WordPress程序

步骤4：配置Apache发布网站

步骤5：初始化安装

步骤6：服务测试





操作视频 / 现场演示



## ✓ 任务5：通过WordPress建设内容网站

### ■ 任务目标：

- WordPress部署成功
- 在本地主机上正常浏览内容网站

无价、且免费  
下载WordPress并用于您的站点。

[下载WordPress 5.7](#)  
下载 .tar.gz

**安装**  
借助我们著名的五分钟安装，首次运行WordPress极为简单。我们为您准备了一份[指南](#)来帮助您完成安装过程。

**版本发布通知**  
想要在WordPress有新版本发布时得到通知？加入[WordPress公告邮件列表](#)，我们在有新的稳定版本发布时会发邮件通知您。

**系统要求**  
我们推荐服务器运行PHP 7.4或更高版本，及MySQL 5.6或MariaDB 10.1或更高版本。我们也推荐Apache或Nginx作为运行WordPress的可靠选项，但这两者都不是必须的。

**更多资源**

- [正式发布版本](#)
- [Beta测试/每日构建版本](#)
- [下载计数器](#)
- [源代码](#)





## 命令指南 / 操作引导

1. #使用php -v命令验证已安装的PHP版本
2. [root@Project-03-Task-03 ~]# php -v
3. #使用rpm -qa命令验证已安装的MariaDB版本
4. [root@Project-03-Task-03 ~]# rpm -qa | grep mariadb
  
5. #使用root账户登录MariaDB数据库
6. [root@Project-03-Task-03 ~]# mysql -u root -p
7. #在MariaDB数据库内进行操作，创建名为wordpressdb的数据库
8. MariaDB [(none)]> create database wordpressdb;
  
9. #使用yum工具安装php-json模块
10. [root@Project-03-Task-03 html]# yum install -y php-json
  
11. #使用yum工具安装wget
12. [root@Project-03-Task-03 ~]# yum install -y wget
13. #使用wget工具下载WordPress文件到指定目录，应用程序存放在账号目录下。
14. [root@Project-03-Task-03 ~]# wget https://cn.wordpress.org/latest-zh\_CN.zip
15. #使用yum工具安装unzip
16. [root@Project-03-Task-03 ~]# yum install -y unzip
17. #使用unzip工具将~/latest-zh\_CN.zip文件解压到/var/www目录下
18. [root@Project-03-Task-03 ~]# unzip latest-zh\_CN.zip -d /var/www
19. #设置wordpress目录所属用户和组均为apache
20. [root@Project-03-Task-03 ~]# chown -R apache:apache /var/www/wordpress
21. #设置wordpress目录的权限为755
22. [root@Project-03-Task-03 ~]# chmod -R 755 /var/www/wordpress





## 命令指南 / 操作引导

1. #编辑apache配置文件, 发布内容网站
2. [root@Project-03-Task-03 ~] vi /etc/httpd/conf/httpd.conf
3. #默认网站配置
4. Listen 80
5. #将默认网站目录/var/www/html, 改为/var/www/wordpress
6. DocumentRoot "/var/www/wordpress"
7. <Directory "/var/www/wordpress">
8.     Options Indexes FollowSymLinks
9.     AllowOverride None
10.     Require all granted
11. </Directory>
  
12. #编辑/etc/httpd/conf.d/welcome.conf, 关闭Apache欢迎信息
13. [root@Project-03-Task-03 ~] vi / etc/httpd/conf.d/welcome.conf
14. # This configuration file enables the default "Welcome" page if there
15. # is no default index page present for the root URL. To disable the
16. # Welcome page, comment out all the lines below.
17. #
18. # NOTE: if this file is removed, it will be restored on upgrades.
19. #<LocationMatch "^/+\$">
20. #为了排版方便此处删除了部分提示信息
21. #</Directory>



## 4. LAMP

### 4.3 任务6

#### 任务6：查看Apache的运行状态

步骤1：安装并启用Apache运行状态模块

步骤2：配置并实现server-status

步骤3：配置并实现server-info

步骤4：查看Apache运行状态信息





操作视频 / 现场演示

## ✓ 任务6：查看Apache的运行状态

### ■ 任务目标：

- 在本地主机上通过浏览器查看Apache Server-Status
- 在本地主机上通过浏览器查看Apache Server-Info





## 命令指南 / 操作引导

1. #查看Apache模块，确定已经安装mod\_status、mod\_info
2. [root@Project-03-Task-02 ~]# ls /etc/httpd/modules/mod\_status\*
3. /etc/httpd/modules/mod\_status.so
4. [root@Project-03-Task-02 ~]# ls /etc/httpd/modules/mod\_info\*
5. /etc/httpd/modules/mod\_info.so
  
6. #查看Apache模块配置信息，确定Apache的mod\_status、mod\_info模块已经启用
7. [root@Project-03-Task-02 ~]# cat /etc/httpd/conf.modules.d/00-base.conf | grep -e mod\_status -e mod\_info
8. LoadModule info\_module modules/mod\_info.so
9. LoadModule status\_module modules/mod\_status.so
10. #查看结果如果显示为#开始，表示该行注释，通过vi编辑该文件将注释删除
  
11. #创建mod\_status、mod\_info的配置文件
12. [root@Project-03-Task-02 ~]# vi /etc/httpd/conf.modules.d/ApacheMonitor.conf
13. #配置mod\_status、mod\_info的访问路径，并设置为本地主机访问
14. <location /server-status>
15.     SetHandler server-status
16.     Require ip 10.10.2.100
17. </location>
  
18. <location /server-info>
19.     SetHandler server-info
20.     Require ip 10.10.2.100
21. </location>
  
22. #使用systemctl reload命令重新载入Apache配置文件
23. [root@Project-03-Task-02 ~]# systemctl reload httpd







## Apache Server Status for 10.10.2.105 (via 10.10.2.105)

Server Version: Apache/2.4.37 (centos)  
Server MPM: event  
Server Built: Dec 23 2019 20:45:34

Current Time: Sunday, 22-Mar-2020 21:57:13 CST  
Restart Time: Sunday, 22-Mar-2020 21:57:07 CST  
Parent Server Config. Generation: 1  
Parent Server MPM Generation: 0  
Server uptime: 5 seconds  
Server load: 0.00 0.00 0.00  
Total accesses: 1 - Total Traffic: 4 kB - Total Duration: 0  
CPU Usage: u.02 s.02 cu0 cs0 - .8% CPU load  
.2 requests/sec - 819 B/second - 4096 B/request - 0 ms/request  
1 requests currently being processed, 74 idle workers

Slot	PID	Stopping	Connections			Threads		Async connections	
			total	accepting	busy	idle	writing	keep-alive	closing
0	7177	no	0	yes	0	25	0	0	0
1	7178	no	0	yes	1	24	0	0	0
2	7179	no	0	yes	0	25	0	0	0
Sum	3	0	0		1	74	0	0	0

M

### Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
"w" Sending Reply, "R" Keepalive (read), "b" DNS Lookup,  
"c" Closing connection, "L" Logging, "G" Gracefully finishing,  
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7178	0/1/1	_	0.00	2	0	0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1	
1-0	7178	0/0/0	W	0.00	0	0	0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1	

- The Status module allows a server administrator to find out how well their server is performing. A HTML page is presented that gives the current server statistics in an easily readable form. If required this page can be made to automatically refresh (given a compatible browser). Another page gives a simple machine-readable list of the current server state.

- The details given are:

- The number of workers serving requests
- The number of idle workers
- The status of each worker, the number of requests that worker has performed and the total number of bytes served by the worker (\*)
- A total number of accesses and byte count served (\*)
- The time the server was started/restarted and the time it has been running for
- Averages giving the number of requests per second, the number of bytes served per second and the average number of bytes per request (\*)
- The current percentage CPU used by each worker and in total by all workers combined (\*)

The current hosts and requests being processed (\*)

Srv Child Server number - generation

PID OS process ID

Acc Number of accesses this connection / this child / this slot

M Mode of operation

CPU CPU usage, number of seconds

SS Seconds since beginning of most recent request

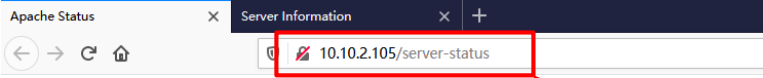
Req Milliseconds required to process most recent request

Dur Sum of milliseconds required to process all requests

Conn Kilobytes transferred this connection

Child Megabytes transferred this child

Slot Total megabytes transferred this slot



# Apache Server Status for 10.10.2.105 (via 10.10.2.105)

Server Version: Apache/2.4.37 (centos)  
Server MPM: event  
Server Built: Dec 23 2019 20:45:34

Current Time: Sunday, 22-Mar-2020 21:57:13 CST  
Restart Time: Sunday, 22-Mar-2020 21:57:07 CST  
Parent Server Config. Generation: 1  
Parent Server MPM Generation: 0  
Server uptime: 5 seconds  
Server load: 0.00 0.00 0.00  
Total accesses: 1 - Total Traffic: 4 kB - Total Duration: 0  
CPU Usage: u.02 s.02 cu0 cs0 -.8% CPU load  
.2 requests/sec - 819 B/second - 4096 B/request - 0 ms/request  
1 requests currently being processed, 74 idle workers

Slot	PID	Stopping	Connections			Threads		Async connections	
			total	accepting	busy	idle	writing	keep-alive	closing
0	7177	no	0	yes	0	25	0	0	0
1	7178	no	0	yes	1	24	0	0	0
2	7179	no	0	yes	0	25	0	0	0
Sum	3	0	0		1	74	0	0	0

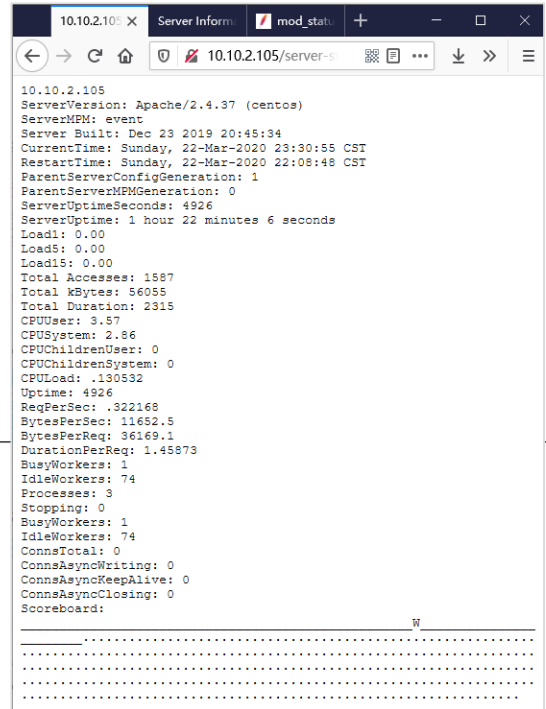
Scoreboard Key:  
" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
"w" Sending Reply, "R" Keepalive (read), "b" DNS Lookup,  
"c" Closing connection, "L" Logging, "G" Gracefully finishing,  
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7178	0/1/1	_	0.00	2	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
1-0	7178	0/0/0	W	0.00	0	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1

- Srv Child Server number - generation
- PID OS process ID
- Acc Number of accesses this connection / this child / this slot
- M Mode of operation
- CPU CPU usage, number of seconds
- SS Seconds since beginning of most recent request
- Req Milliseconds required to process most recent request
- Dur Sum of milliseconds required to process all requests
- Conn Kilobytes transferred this connection
- Child Megabytes transferred this child
- Slot Total megabytes transferred this slot

## Apache Server Status三种访问方式:

- 常规方式: <http://your.server.name/server-status>
- 自动刷新: <http://your.server.name/server-status?refresh=N>
- 机器阅读: <http://your.server.name/server-status?auto>



## Apache Server Status for 10.10.2.105 (via 10.10.2.105)

```
Server Version: Apache/2.4.37 (centos)
Server MPM: event
Server Built: Dec 23 2019 20:45:34
```

```
Current Time: Sunday, 22-Mar-2020 21:57:13 CST
Restart Time: Sunday, 22-Mar-2020 21:57:07 CST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 seconds
Server load: 0.00 0.00 0.00
```

```
Total accesses: 1 - Total Traffic: 4 kB - Total Duration: 0
CPU Usage: u.02 s.02 cu0 cs0 - .8% CPU load
.2 requests/sec - 819 B/second - 4096 B/request - 0 ms/request
1 requests currently being processed, 74 idle workers
```

Slot	PID	Stopping	Connections			Threads			Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing		
0	7177	no	0	yes	0	25	0	0	0	0	
1	7178	no	0	yes	1	24	0	0	0	0	
2	7179	no	0	yes	0	25	0	0	0	0	
Sum	3	0	0		1	74	0	0	0	0	

### Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "w" Sending Reply, "R" Keepalive (read), "b" DNS Lookup,  
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7178	0/1/1	_	0.00	2	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
1-0	7178	0/0/0	W	0.00	0	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1

- Server Version Apache Http Server 软件版本
- Server MPM Apache 的工作模式
- Server Built Apache 当前版本的编译时间
- Current Time 当前系统时间
- Restart Time 最近重启httpd服务的时间
- Parent Server Config. Generation httpd的父进程重启的次数
- Parent Server MPM Generation httpd的event模式中主进程重启的次数
- Server uptime 最近启动httpd到现在共计运行的时长
- Server load 服务器负载, 分别是1min 5min 15min时间范围内

**Srv** Child Server number - generation

**PID** OS process ID

**Acc** Number of accesses this connection / this child / this slot

**M** Mode of operation

**CPU** CPU usage, number of seconds

**SS** Seconds since beginning of most recent request

**Req** Milliseconds required to process most recent request

**Dur** Sum of milliseconds required to process all requests

**Conn** Kilobytes transferred this connection

**Child** Megabytes transferred this child

**Slot** Total megabytes transferred this slot

## Apache Server Status for 10.10.2.105 (via 10.10.2.105)

Server Version: Apache/2.4.37 (centos)  
 Server MPM: event  
 Server Built: Dec 23 2019 20:45:34

Current Time: Sunday, 22-Mar-2020 21:57:13 CST  
 Restart Time: Sunday, 22-Mar-2020 21:57:07 CST  
 Parent Server Config. Generation: 1  
 Parent Server MPM Generation: 0  
 Server uptime: 5 seconds  
 Server load: 0.00 0.00 0.00

Total accesses: 1 - Total Traffic: 4 kB - Total Duration: 0  
 CPU Usage: u.02 s.02 cu0 cs0 - .8% CPU load  
 .2 requests/sec - 819 B/second - 4096 B/request - 0 ms/request  
 1 requests currently being processed, 74 idle workers

Slot	PID	Stopping	Connections			Threads			Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing		
0	7177	no	0	yes	0	25	0	0	0	0	
1	7178	no	0	yes	1	24	0	0	0	0	
2	7179	no	0	yes	0	25	0	0	0	0	
Sum	3	0	0		1	74	0	0	0	0	

### Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "w" Sending Reply, "R" Keepalive (read), "b" DNS Lookup,  
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7178	0/1/1_	0	0.00	2	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
1-0	7178	0/0/0	W	0.00	0	0	0	0.00	0.00	10.10.2.100	http/1.1		fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1

- Total accesses httpd启动后接收的请求总数
- Total Traffic httpd启动后传送的数据总量
- Total Duration httpd启动后使用CPU的总时长

- CPU Usage CPU使用情况

- .015 requests/sec - 91 B/second - 5.9 kB/request - .35 ms/request  
httpd启动后的平均负载

每秒多少个请求  
 每秒传送多少数据  
 每个请求平均传送多少数据  
 每个请求平均耗时多长

- 1 requests currently being processed, 74 idle workers

当前占用的线程数  
 当前空闲的线程数

Srv Child Server number - generation

PID OS process ID

Acc Number of accesses this connection / this child / this slot

M Mode of operation

CPU CPU usage, number of seconds

SS Seconds since beginning of most recent request

Req Milliseconds required to process most recent request

Dur Sum of milliseconds required to process all requests

Conn Kilobytes transferred this connection

Child Megabytes transferred this child

Slot Total megabytes transferred this slot

Apache Status - Server Information

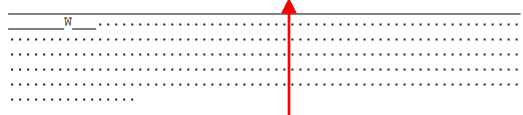
10.10.2.105/server-status

## Apache Server Status for 10.10.2.105 (via 10.10.2.105)

Server Version: Apache/2.4.37 (centos)  
 Server MPM: event  
 Server Built: Dec 23 2019 20:45:34

Current Time: Sunday, 22-Mar-2020 22:43:41 CST  
 Restart Time: Sunday, 22-Mar-2020 22:08:48 CST  
 Parent Server Config. Generation: 1  
 Parent Server MPM Generation: 0  
 Server uptime: 34 minutes 52 seconds  
 Server load: 0.00 0.00 0.00  
 Total accesses: 28 - Total Traffic: 163 kB - Total Duration: 9  
 CPU Usage: u.22 s.49 cu0 cs0 - .0339% CPU load  
 .0134 requests/sec - 79 B/second - 5.8 kB/request - .321429 ms/request  
 1 requests currently being processed, 74 idle workers

Slot	PID	Stopping	Connections		Threads		Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing
0	7629	no	0	yes	0	25	0	0	0
1	7630	no	0	yes	0	25	0	0	0
2	7631	no	0	yes	1	24	0	0	0
Sum	3	0	0		1	74	0	0	0



Scoreboard Key:  
 " " Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "W" Sending Reply, "k" Keepalive (read), "b" DNS Lookup,  
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7630	0/1/1	_	0.00	2088	0	0.0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
1-0	7630	0/1/1	_	0.00	2088	0	0.0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.23	1253	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.08	256	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2	_	0.24	1214	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.39	760	0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.51	395	0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.24	1222	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.08	760	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.39	760	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.56	255	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.56	255	0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.00	1206	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.23	255	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.08	395	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2	_	0.56	256	0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1

```
[root@VM-Project-01-Task-02 ~]# ps aux | grep httpd
root      7626  0.0  1.3 280268 11336 ?        Ss   22:08   0:00 /usr/sbin/httpd -DFO
apache    7628  0.0  1.0 292484  8552 ?        S    22:08   0:00 /usr/sbin/httpd -DFO
apache    7629  0.0  1.4 1350256 11944 ?        S1   22:08   0:00 /usr/sbin/httpd -DFOREGROUND
apache    7630  0.0  1.6 1481384 13992 ?        S1   22:08   0:00 /usr/sbin/httpd -DFOREGROUND
apache    7631  0.0  1.4 1350256 11944 ?        S1   22:08   0:00 /usr/sbin/httpd -DFOREGROUND
root      7880  0.0  0.1 12320 1084 pts/0    S+   22:44   0:00 grep --color=auto httpd

[root@VM-Project-01-Task-02 ~]#
[root@VM-Project-01-Task-02 ~]#
```

## Apache Server Status for 10.10.2.105 (via 10.10.2.105)

Server Version: Apache/2.4.37 (centos)  
Server MPM: event  
Server Built: Dec 23 2019 20:45:34

Current Time: Sunday, 22-Mar-2020 22:43:41 CST  
Restart Time: Sunday, 22-Mar-2020 22:08:48 CST  
Parent Server Config. Generation: 1  
Parent Server MPM Generation: 0  
Server uptime: 34 minutes 52 seconds  
Server load: 0.00 0.00 0.00  
Total accesses: 28 - Total Traffic: 163 kB - Total Duration: 9  
CPU Usage: u.22 s.49 cu0 cs0 - .0339% CPU load  
.0134 requests/sec - 79 B/second - 5.8 kB/request - .321429 ms/request  
1 requests currently being processed, 74 idle workers

Slot	PID	Stopping	Connections		Threads		Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing
0	7629	no	0	yes	0	25	0	0	0
1	7630	no	0	yes	0	25	0	0	0
2	7631	no	0	yes	1	24	0	0	0
Sum	3	0	0		1	74	0	0	0

```
.....
.....
.....
.....
.....
.....
.....
.....
```

### Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
"W" Sending Reply, "k" Keepalive (read), "D" DNS Lookup,  
"C" Closing connection, "L" Logging, "G" Gracefully finishing,  
"I" Idle cleanup of worker, "." Open slot with no current process

### • Scoreboard Key:

- " " Waiting for Connection 等待连接中
- "s" Starting up 启动中
- "r" Reading Request 正在读取请求
- "W" Sending Reply 正在发送响应
- "k" Keepalive (read) 保持联机的状态
- "D" DNS Lookup 正在查找DNS
- "C" Closing connection 正在关闭连接
- "L" Logging 正在写入日志
- "G" Gracefully finishing 进入正常结束程序
- "I" Idle cleanup of worker 处理限制
- "." Open slot with no current process 没有创建该线程

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7630	0/1/1_	0.00	2088	0	0.0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
1-0	7630	0/1/1_	0.00	2088	0	0.0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.23	1253	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.08	256	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2_	0.24	1214	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.39	760	0	0.0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1_	0.51	395	0	0.0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1_	0.24	1222	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.08	760	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.39	760	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.56	255	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.56	255	0	0.0	0.0	0.00	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1_	0.00	1206	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.23	255	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1_	0.08	395	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2_	0.56	256	0	0.0	0.0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1

Sum	3	0	0	1	74	0	0	0
...	...	...	...	...	...	...	...	...

## Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "W" Sending Reply, "k" Keepalive (read), "b" DNS Lookup,  
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	7630	0/1/1	_	0.00	2088	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
1-0	7630	0/1/1	_	0.00	2088	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.23	1253	0	0	0.01	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.08	256	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1		
2-0	7631	0/2/2	_	0.24	1214	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.39	760	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.51	395	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.24	1222	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.08	760	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1		
2-0	7631	0/1/1	_	0.39	760	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.56	255	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.56	255	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1
2-0	7631	0/1/1	_	0.00	1206	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.23	255	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1		
2-0	7631	0/1/1	_	0.08	395	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1		
2-0	7631	0/2/2	_	0.56	256	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2	_	0.24	1211	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/2/2	_	0.50	395	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.24	1223	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.00	1217	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	W	0.23	0	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.00	1206	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.00	1217	0	0	0.0	0.01	0.01	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /server-status HTTP/1.1
2-0	7631	0/1/1	_	0.56	256	0	0	0.0	0.00	0.00	10.10.2.100	http/1.1	fe80::9a67:79d2:17fa:ce6b:80	GET /favicon.ico HTTP/1.1

- Srv Child Server number - generation
- PID OS process ID 处理该请求的进程的PID
- Acc Number of accesses this connection / this child / this slot  
当前连接 / 当前连接所对应的子进程 / 当前连接所对应的子进程所属槽的读取次数
- M Mode of operation 目前的状态
- CPU CPU usage, number of seconds 使用的CPU资源
- SS Seconds since beginning of most recent request  
距离上次处理要求的时长 (多少秒)
- Req Milliseconds required to process most recent request  
最后处理操作所耗费的时间 (多少毫秒)
- Dur Sum of milliseconds required to process all requests
- Conn Kilobytes transferred this connection  
当前连接传送的数据量
- Child Megabytes transferred this child  
当前连接所对应的子进程传送的数据量
- Slot Total megabytes transferred this slot  
当前连接所对应的子进程所属槽传送的数据量
- Client 客户端IP地址
- Protocol 使用的协议
- VHost Apache服务器的IP地址和接收请求的端口
- Request 请求的路径, 使用的协议

Srv Child Server number - generation  
 PID OS process ID  
 Acc Number of accesses this connection / this child / this slot  
 M Mode of operation  
 CPU CPU usage, number of seconds  
 SS Seconds since beginning of most recent request  
 Req Milliseconds required to process most recent request  
 Dur Sum of milliseconds required to process all requests  
 Conn Kilobytes transferred this connection  
 Child Megabytes transferred this child  
 Slot Total megabytes transferred this slot

- A connection is a link between a client and a child which resides in a slot.
- A child is a process or a thread that handles requests.
- A slot is a placeholder for a child. There may or may not be a child there depending on max/min spare servers.

10.10.2.105/server-info

## Apache Server Information

Subpages:  
[Configuration Files](#), [Server Settings](#), [Module List](#), [Active Hooks](#), [Available Providers](#)

Sections:  
[Loaded Modules](#), [Server Settings](#), [Startup Hooks](#), [Request Hooks](#), [Other Hooks](#), [Providers](#)

### Loaded Modules

core.c, event.c, http core.c, mod access.compat.c, mod actions.c, mod alias.c, mod allowmethods.c, mod auth.basic.c, mod auth.digest.c, mod authn.anon.c, mod authn.core.c, mod authn.dbd.c, mod authn.dbm.c, mod authn.file.c, mod authn.socache.c, mod authz.core.c, mod authz.dbd.c, mod authz.dbm.c, mod authz.davfile.c, mod authz.host.c, mod authz.owner.c, mod authz.user.c, mod autoindex.c, mod cgi.c, mod data.c, mod dav.c, mod dav.fs.c, mod dav.lock.c, mod dbd.c, mod deflate.c, mod dir.c, mod dumpio.c, mod echo.c, mod env.c, mod expires.c, mod ext\_filter.c, mod info.c, mod lbmethod.byrequests.c, mod lbmethod.bytraffic.c, mod lbmethod.heartbeat.c, mod log.config.c, mod logio.c, mod lua.c, mod mod\_proxy ajp.c, mod proxy balancer.c, mod proxy connect.c, mod proxy express.c, mod proxy fcgi.c, mod proxy fdpass.c, mod proxy ftp.c, mod proxy hcheck.c, mod proxy wstunnel.c, mod remotel.c, mod reqtimeout.c, mod request.c, mod rewrite.c, mod setenvif.c, mod slotmem.plain.c, mod slotmem.shm.c, mod socache.shm.c, mod substitute.c, mod suexec.c, mod systemd.c, mod unique.id.c, mod unixd.c, mod userdir.c, mod version.c, mod vhost.alias.c, mod watchdog.c.

### Server Settings

Server Version: Apache/2.4.37 (centos)  
 Server Built: Dec 23 2019 20:45:34  
 Server loaded APR Version: 1.6.3  
 Compiled with APR Version: 1.6.3  
 Server loaded APU Version: 1.6.1  
 Compiled with APU Version: 1.6.1  
 Module Magic Number: 20120211:83  
 Hostname/port: 10.10.2.105:80  
 Timeouts: connection: 60 keep-alive: 5  
 MPM Name: event  
 MPM Information: Max Daemons: 3 Threaded: yes Forked: yes  
 Server Architecture: 64-bit  
 Server Root: /etc/httpd  
 Config File: /etc/httpd/conf/httpd.conf  
 Server Built With:  
 -D APR\_HAS\_SENDFILE  
 -D APR\_HAS\_MMAP  
 -D APR\_HAVE\_IPV6 (IPv4-mapped addresses enabled)  
 -D APR\_USE\_SYSVSEM\_SERIALIZE  
 -D SINGLE\_LISTEN\_UNSERIALIZED\_ACCEPT  
 -D APR\_HAS\_OTHER\_CHILD  
 -D AP\_HAVE\_RELIABLE\_PIPED\_LOGS  
 -D HTTPD\_ROOT="/etc/httpd"  
 -D SUEXEC\_BIN="/usr/sbin/suexec"  
 -D DEFAULT\_PIDLOG="run/httpd.pid"  
 -D DEFAULT\_SCOREBOARD="logs/apache\_runtime\_status"  
 -D DEFAULT\_ERRORLOG="logs/error\_log"  
 -D AP\_TYPES\_CONFIG\_FILE="conf/mime.types"  
 -D SERVER\_CONFIG\_FILE="conf/httpd.conf"

### Startup Hooks

Pre-Config:  
 -10 core.c  
 -10 mod\_log\_config.c  
 -10 mod\_logio.c

### Apache Server Information

Configuration:

```
In file: /etc/httpd/conf/httpd.conf
45: Listen 80
In file: /etc/httpd/conf.modules.d/000/apache2mon.conf
1: <Location /server-status>
2:   SetHandler server-status
3:   Require ip 10.10.2.100
   </Location>
6: <Location /server-info>
7:   SetHandler server-info
8:   Require ip 10.10.2.100
   </Location>
In file: /etc/httpd/conf/httpd.conf
69: User apache
70: Group apache
89: ServerAdmin root@localhost
105: <Directory />
106:   AllowOverride none
107:   Require all denied
   </Directory>
122: DocumentRoot "/var/www/wordpress"
127: <Directory "/var/www/wordpress">
128:   Options Indexes FollowSymLinks
129:   AllowOverride None
131:   Require all granted
   </Directory>
135: <Directory "/var/www/html">
148:   Options Indexes FollowSymLinks
155:   AllowOverride None
160:   Require all granted
   </Directory>
```

Apache Server Information两种访问方式:

- 常规方式: <http://your.server.name/server-info>
- 配置阅读: <http://your.server.name/server-status?config>



## Apache Server Information

Subpages:

[Configuration Files](#), [Server Settings](#), [Module List](#), [Active Hooks](#), [Available Providers](#)

Sections:

[Loaded Modules](#), [Server Settings](#), [Startup Hooks](#), [Request Hooks](#), [Other Hooks](#), [Providers](#)

导航

### Loaded Modules

模块导航

[core.c](#), [event.c](#), [http\\_core.c](#), [mod\\_access\\_compat.c](#), [mod\\_actions.c](#), [mod\\_alias.c](#), [mod\\_allowmethods.c](#), [mod\\_auth\\_basic.c](#), [mod\\_auth\\_digest.c](#), [mod\\_authn\\_anon.c](#), [mod\\_authn\\_core.c](#), [mod\\_authn\\_dbd.c](#), [mod\\_authn\\_dbm.c](#), [mod\\_authn\\_file.c](#), [mod\\_authn\\_socache.c](#), [mod\\_authz\\_core.c](#), [mod\\_authz\\_dbm.c](#), [mod\\_authz\\_groupfile.c](#), [mod\\_authz\\_host.c](#), [mod\\_authz\\_owner.c](#), [mod\\_authz\\_user.c](#), [mod\\_autoindex.c](#), [mod\\_brotli.c](#), [mod\\_cache.c](#), [mod\\_cache\\_disk.c](#), [mod\\_cache\\_socache.c](#), [mod\\_cgid.c](#), [mod\\_data.c](#), [mod\\_dav.c](#), [mod\\_dav\\_fs.c](#), [mod\\_dav\\_lock.c](#), [mod\\_dbd.c](#), [mod\\_deflate.c](#), [mod\\_dir.c](#), [mod\\_dumpio.c](#), [mod\\_echo.c](#), [mod\\_env.c](#), [mod\\_expires.c](#), [mod\\_ext\\_filter.c](#), [mod\\_filter.c](#), [mod\\_headers.c](#), [mod\\_http2.c](#), [mod\\_include.c](#), [mod\\_info.c](#), [mod\\_lbmethod\\_bybusyness.c](#), [mod\\_lbmethod\\_byrequests.c](#), [mod\\_lbmethod\\_bytraffic.c](#), [mod\\_lbmethod\\_heartbeat.c](#), [mod\\_log\\_config.c](#), [mod\\_logio.c](#), [mod\\_lua.c](#), [mod\\_macro.c](#), [mod\\_mime.c](#), [mod\\_mime\\_magic.c](#), [mod\\_negotiation.c](#), [mod\\_proxy.c](#), [mod\\_proxy\\_ajp.c](#), [mod\\_proxy\\_balancer.c](#), [mod\\_proxy\\_connect.c](#), [mod\\_proxy\\_express.c](#), [mod\\_proxy\\_fcgi.c](#), [mod\\_proxy\\_fdpass.c](#), [mod\\_proxy\\_ftp.c](#), [mod\\_proxy\\_hcheck.c](#), [mod\\_proxy\\_http.c](#), [mod\\_proxy\\_http2.c](#), [mod\\_proxy\\_scgi.c](#), [mod\\_proxy\\_wsgi.c](#), [mod\\_proxy\\_wstunnel.c](#), [mod\\_remotepip.c](#), [mod\\_request.c](#), [mod\\_rewrite.c](#), [mod\\_setenvif.c](#), [mod\\_slotmem\\_plain.c](#), [mod\\_slotmem\\_shm.c](#), [mod\\_soc.c](#), [mod\\_socache\\_dbm.c](#), [mod\\_socache\\_memcache.c](#), [mod\\_socache\\_shmcb.c](#), [mod\\_status.c](#), [mod\\_substitute.c](#), [mod\\_suexec.c](#), [mod\\_systemd.c](#), [mod\\_unique\\_id.c](#), [mod\\_unixd.c](#), [mod\\_userdir.c](#), [mod\\_version.c](#), [mod\\_vhost\\_alias.c](#), [mod\\_watchdog.c](#).

### Server Settings

**Server Version:** Apache/2.4.37 (centos)

**Server Built:** Dec 23 2019 20:45:34

**Server loaded APR Version:** 1.6.3

**Compiled with APR Version:** 1.6.3

**Server loaded APU Version:** 1.6.1

**Compiled with APU Version:** 1.6.1

**Module Magic Number:** 20120211:83

**Hostname/port:** 10.10.2.105:80

**Timeouts:** connection: 60 keep-alive: 5

**MPM Name:** event

**MPM Information:** Max Daemons: 3 Threaded: yes Forked: yes

**Server Architecture:** 64-bit

**Server Root:** /etc/httpd

**Config File:** /etc/httpd/conf/httpd.conf

**Server Built With:**

```
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D HTTPD_ROOT="/etc/httpd"
-D SUEXEC_BIN="/usr/sbin/suexec"
-D DEFAULT_PIDLOG="run/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

### Startup Hooks

**Pre-Config:**

```
-10 core.c
-10 mod_log_config.c
-10 mod_logio.c
```

## 5. 提升Apache的安全性

- 网站安全是网络安全和信息安全的重要组成部分，提升Apache网站服务器的安全性是保障网站安全的重要措施。
- Apache提供了多个手段以提升Apache的安全性，保障网站安全可靠提供服务。
  - 设置网站访问范围
  - 隐藏服务器敏感信息
  - 禁止网站目录浏览
  - 开启SELinux、防火墙进行安全防护



## 5. 提升Apache的安全性

### 5.1 设置网站访问范围

- 设置网站访问范围可以有效阻隔恶意主机攻击，极大提升网站安全性。
- Apache服务器通过Require选项实现网站访问范围限制，可通过修改配置文件实现，并支持针对网站、虚拟目录自由进行定义。
- 示例：将网站的可访问范围设置为两条规则：
  - 允许所有地址访问内容网站
  - 禁止10.10.2.116地址访问内容网站



## 5. 提升Apache的安全性

### 5.1 设置网站访问范围

#### 1. 使用vi工具编辑网站配置文件，配置文件信息如下所示。

1. <Directory "/var/www/wordpress">
2.     Options Indexes FollowSymLinks
3.     AllowOverride None
4.     #设置网站访问范围
5.     <RequireAll>
6.         Require all granted
7.         Require not ip 10.10.2.116
8.     </RequireAll>
9. </Directory>

#### 2. 配置完成后，重新载入配置文件使其生效。

1. systemctl reload httpd



## 5. 提升Apache的安全性

### 5.1 设置网站访问范围

#### 1. Apache的Require项常用指令。

1. Require all granted, 允许所有来源访问;
2. Require all denied, 拒绝所有来源访问;
3. Require ip 127.0.0.1, 只允许特定IP段访问, 多个IP段之间使用空格隔开, 这里是只允许IP地址为127.0.0.1的来源主机访问;
4. Require host domain.com, 只允许来自域名domain.com的主机访问;
5. Require项可以配置多个。

#### 2. Require项配合<RequireAll>、<RequireAny>、<RequireNone>标签对可以进行更加复杂的访问限制。

1. RequireAll, 访问请求必须全部符合设置的允许访问规则, 才能访问网站;
2. RequireAny, 访问请求符合设置的任意一条允许访问规则, 就能访问网站;
3. RequireNone, 访问请求符合设置的任意一条规则, 都不能访问网站, 不能独立使用, 一般与其它标签对配合使用。



## 5. 提升Apache的安全性

### 5.2 隐藏服务器敏感信息

- 知己知彼方可百战百胜，隐藏Apache网站服务器和PHP解析器的敏感信息，亦可有效降低精准攻击的概率，降低服务器的风险。
- Apache网站服务器通过ServerTokens选项隐藏版本等敏感信息。
- PHP解析器通过expose\_php选项隐藏敏感信息。
  
- 示例：实现Apache网站服务器和PHP解析器的信息保护。
  - 修改httpd.conf文件隐藏网站服务器敏感信息
  - 修改php.ini文件隐藏PHP敏感信息



## 5. 提升Apache的安全性

### 5.2 隐藏服务器敏感信息

1. 使用vi工具编辑Apache配置文件/etc/httpd/conf/httpd.conf，配置文件信息如下所示。
  1. #在文件的最底部增加下述内容
  2. ServerTokens Prod
2. 使用vi工具编辑php配置文件/etc/php.ini，编辑后的配置文件信息如下所示。
  1. #将expose\_php = On改为expose\_php = Off
  2. expose\_php = Off
3. 配置完成后，重新载入配置文件使其生效。
  1. systemctl reload httpd
  2. systemctl reload php-fpm



消息头 Cookie 参数 响应 耗时 堆栈跟踪

状态码: 200 OK ?

版本: HTTP/1.1 编辑和重发

过滤消息头

▼ 响应头 (310 字节) 原始头

- Connection: Keep-Alive
- Content-Type: text/html; charset=UTF-8
- Date: Thu, 20 Feb 2020 11:58:13 GMT
- Keep-Alive: timeout=5, max=100
- Link: <http://10.10.2.105/index.php/...on/>; rel="https://api.w.org/"
- Server: Apache/2.4.37 (centos)
- Transfer-Encoding: chunked
- X-Powered-By: PHP/7.3.5

消息头 Cookie 参数 响应 耗时 堆栈跟踪

状态码: 200 OK ?

版本: HTTP/1.1 编辑和重发

过滤消息头

▼ 响应头 (269 字节) 原始头

- Connection: Keep-Alive
- Content-Type: text/html; charset=UTF-8
- Date: Thu, 20 Feb 2020 12:00:09 GMT
- Keep-Alive: timeout=5, max=100
- Link: <http://10.10.2.105/index.php/...on/>; rel="https://api.w.org/"
- Server: Apache
- Transfer-Encoding: chunked





# 5. 提升Apache的安全性

## 5.2 隐藏服务器敏感信息

1. Apache的ServerTokens共有6个选项，其作用分别如下所示。
  1. ServerTokens Full，显示全部信息包含Apache支持的模块及模块版本号
  2. ServerTokens Prod，仅显示网站服务器名称，即Server: Apache
  3. ServerTokens Major，显示网站服务器信息包括主版本号，即Server: Apache/2
  4. ServerTokens Minor，显示网站服务器信息包括次版本号，即Server: Apache/2.4
  5. ServerTokens Min，显示网站服务器信息包含完整版本号，即Server: Apache/2.4.37
  6. ServerTokens OS，显示网站服务器信息包含操作系统类型，即：Server: Apache/2.4.37(centos)
2. PHP的expose\_php共有2个选项，其作用分别如下所示。
  1. On，在网站服务器上显示已安装PHP信息
  2. Off，在网站服务器上不显示已安装PHP信息



## 5. 提升Apache的安全性

### 5.3 禁止网站目录浏览

- 禁止网站目录浏览可有效保护网站信息不被泄露，屏蔽非法用户的恶意浏览。
- 示例：禁止浏览网站目录列表。
  - 修改/etc/httpd/conf/httpd.conf配置文件禁止网站目录浏览
  - 可以针对任何一个网站、虚拟目录进行配置

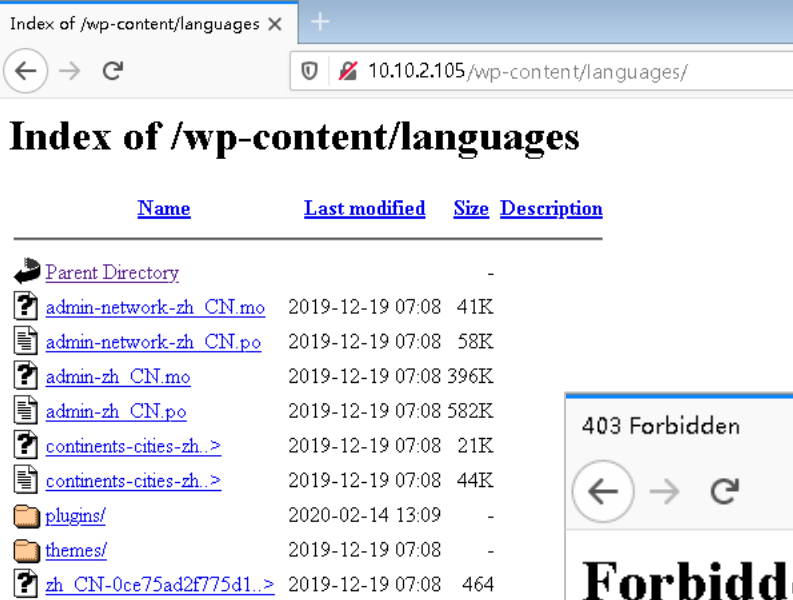


## 5. 提升Apache的安全性

### 5.3 禁止网站目录浏览

1. 使用vi工具编辑网站配置文件，配置文件信息如下所示。
  1. <Directory "/var/www/wordpress">
  2. #Options项设置为None，目录不启用任何服务器特性
  3. **Options None**
  4. AllowOverride None
  5. #设置网站访问范围
  6. <RequireAll>
  7.     Require all granted
  8.     Require not ip 10.10.3.226
  9. </RequireAll>
  10. </Directory>
2. 配置完成后，重新载入配置文件使其生效。
  1. systemctl reload httpd



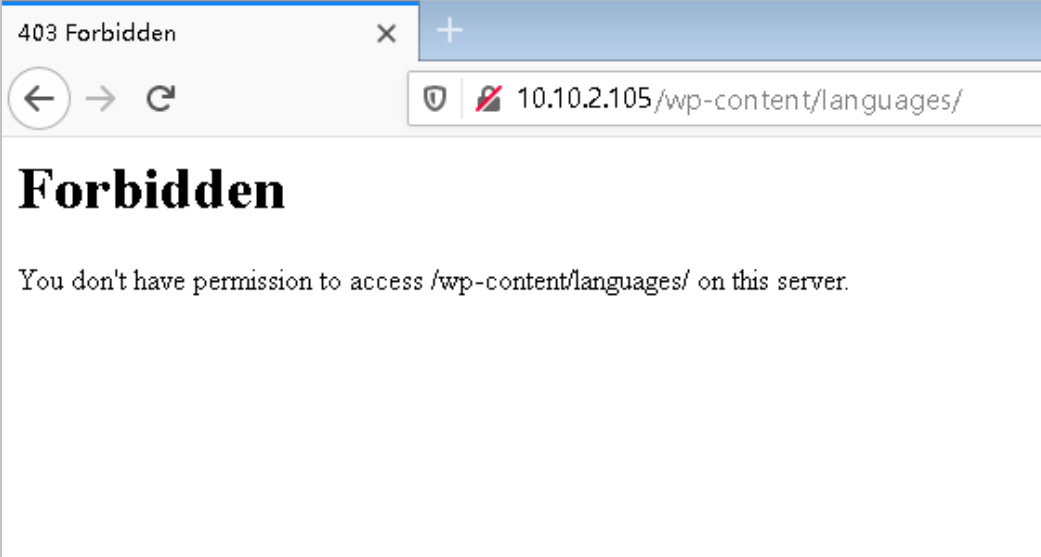


Index of /wp-content/languages/ X +

10.10.2.105/wp-content/languages/

## Index of /wp-content/languages

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#">admin-network-zh_CN.mo</a>	2019-12-19 07:08	41K	
<a href="#">admin-network-zh_CN.po</a>	2019-12-19 07:08	58K	
<a href="#">admin-zh_CN.mo</a>	2019-12-19 07:08	396K	
<a href="#">admin-zh_CN.po</a>	2019-12-19 07:08	582K	
<a href="#">continents-cities-zh_&gt;</a>	2019-12-19 07:08	21K	
<a href="#">continents-cities-zh_&gt;</a>	2019-12-19 07:08	44K	
<a href="#">plugins/</a>	2020-02-14 13:09	-	
<a href="#">themes/</a>	2019-12-19 07:08	-	
<a href="#">zh_CN-0ce75ad2f775d1..&gt;</a>	2019-12-19 07:08	464	



403 Forbidden X +

10.10.2.105/wp-content/languages/

## Forbidden

You don't have permission to access /wp-content/languages/ on this server.



## 5. 提升Apache的安全性

### 5.3 禁止网站目录浏览

#### 1. Apache的Option常用选项如下所示。

1. Options All, 显示除MultiViews之外的所有特性
2. Options MultiViews, 允许多重内容被浏览
3. Options Indexes, 如目录下无index文件, 则显示该目录下的文件
4. Options IncludesNOEXEC, 允许使用服务器端include, 但不可使用#exec和#include功能
5. Options Includes, 允许使用服务器端include
6. Options FollowSymLinks, 在目录中服务器将跟踪符号链接
7. Options SymLinksIfOwnerMatch, 在目录中仅跟踪本站点内的链接
8. Options ExecCGI, 在目录下准许使用CGI

#### 2. Options后可附加多种服务器特性, 特性之间使用空格隔开。



## 5. 提升Apache的安全性

### 5.4 开启SELinux、防火墙进行安全防护

- 前述任务中，均临时关闭了SELinux、防火墙等安全防护措施。
- 为了确保操作系统的安全以及网站访问的安全性，正式服务的业务必须要开启SELinux、防火墙进行安全防护。
- 示例：使用SELinux和Firewalld保护业务安全
  - 将SELinux工作模式设置为enforcing
  - 开启防火墙并开放80端口



## 5. 提升Apache的安全性

### 5.4 开启SELinux、防火墙进行安全防护

1. `[root@Project-03-Task-03 ~]# setenforce 1`
2. `[root@Project-03-Task-03 ~]# sestatus`
  1. #SELinux已开启
  2. SELinux status: enabled
  3. SELinuxfs mount: /sys/fs/selinux
  4. SELinux root directory: /etc/selinux
  5. Loaded policy name: targeted
  6. #当前工作模式为enforcing强制模式
  7. Current mode: enforcing
  8. Mode from config file: enforcing
  9. Policy MLS status: enabled
  10. Policy deny\_unknown status: allowed
  11. Memory protection checking: actual (secure)
  12. Max kernel policy version: 31



## 5. 提升Apache的安全性

### 5.4 开启SELinux、防火墙进行安全防护

1. `[root@Project-03-Task-03 ~]# systemctl start firewalld`
2. `[root@Project-03-Task-03 ~]# systemctl status firewalld`
  1. ● firewalld.service - firewalld - dynamic firewall daemon
  2. Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor p>
  3. #防火墙状态, 结果值为active表示活跃;
  4. Active: **active (running)** since Fri 2020-02-14 16:11:13 CST; 22min ago
  5. Docs: man:firewalld(1)
3. #在防火墙上开放80端口, 并重载防火墙配置使其生效
  1. `[root@Project-03-Task-03 ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent`
  2. `[root@Project-03-Task-03 ~]# firewall-cmd --reload`





## 6.Apache日志分析

### 6.1 Apache日志服务

- Apache日志文件记录了Apache运行历史，通过管理和分析日志可及时了解Apache的运行状态。
- Apache包含访问日志和错误日志两个部分。
  - 日志文件在CentOS中的存放位置是/var/log/httpd/目录
  - 访问日志的文件名为access\_log
  - 错误日志的文件名为error\_log
- 如果使用SSL服务，日志文件将包括关于SSL运行的日志文件。
  - SSL服务的日志文件：ssl\_access\_log、ssl\_error\_log、ssl\_request\_log



## 6.Apache日志分析

### 6.1 Apache日志服务

- Apache的日志记录过程



# 6.Apache日志分析

## 6.1 Apache日志服务

□ Apache日志可通过/etc/httpd/conf/httpd.conf文件进行设置。

```
1.  ErrorLog "logs/error_log"          #错误日志存放位置
2.  LogLevel warn                    #错误日志记录等级
3.  #访问日志的配置信息，通过日志格式字符串可定义访问日志记录的字段。
4.  <IfModule log_config_module>
5.      #定义了名为"combined"的日志记录格式
6.      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
7.      #定义了名为"common"的日志记录格式
8.      LogFormat "%h %l %u %t \"%r\" %>s %b" common
9.  <IfModule logio_module>
10.     #定义记录每个请求输入和输出字节的日志格式，其名称为combinedio
11.     LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %l %O" combinedio
12. </IfModule>
13. #访问日志存放在logs/access_log目录下，日志记录格式为combined定义的格式
14. CustomLog "logs/access_log" combined
15. </IfModule>
```



表 3-2-3 错误日志记录等级表

等级	说明
emerg	紧急，系统无法使用
alert	必须立即采取措施
crit	关键错误，危险情况的警告，由于配置不当所致
error	一般错误
warn	警告信息，不算是错误信息，主要记录服务器出现的某种信息
notice	需要引起注意的情况
info	值得报告的一般消息，比如服务器重启
debug	由运行 debug 模式的程序所产生的消息



表 3-2-2 日志格式中常用字符串含义表

变量	含义
%%	百分号
%a	请求客户端的 IP 地址
%A	本机 IP 地址
%B	不包含 HTTP 头的已发送字节数
%b	不包含 HTTP 头的 CLF 格式的已发送字节数量。当没有发送数据时，显示“-”而不是 0
%D	服务器处理本请求所用时间，单位为微秒
%f	文件名
%h	远端主机
%H	请求使用的协议
%l	远程登录名
%m	请求的方法
%{VARNAME}C	在请求中传送给服务端的 cookie VARNAME 的内容
%{VARNAME}e	环境变量 VARNAME 的值
%{VARNAME}i	发送到服务器的请求头 VARNAME 的内容
%{VARNAME}n	其它模块注释 VARNAME 的内容
%{VARNAME}o	应答头 VARNAME 的内容



表 3-2-2 日志格式中常用字符串含义表

变量	含义
%p	服务器响应请求时使用的端口
%P	响应请求的子进程 ID
%q	查询字符串（如果存在查询字符串，则包含“?”后面的部分；否则，它是一个空字符串）
%r	请求的第一行
%s	状态。对于内部重定向的请求，这里指原来请求的状态。如果用%...>s，则是指后来的请求
%t	接收请求的时间，如：18/Sep/2019:19:18:28 -0400
%{format}t	以指定格式 format 表示的时间
%T	为响应请求而耗费的时间，单位为秒
%u	远程用户
%U	用户所请求的 URL 路径
%v	响应请求的服务器的 ServerName
%V	依照 UseCanonicalName 设置得到的服务器名字
%l	接收的字节数，包含头与正文
%O	发送的字节数，包含头与正文



## 6.Apache日志分析

### 6.2 任务7

任务7：使用CentOS命令分析Apache日志

任务8：使用ApacheTop工具实时分析Apache日志



# 6.Apache日志分析

## 6.2 任务7

### 任务7：使用CentOS命令分析Apache日志

步骤1：对Apache日志内容进行分析

步骤2：明确日志分析需求

步骤3：分析解决思路

步骤4：使用awk、sort、uniq实现





head /var/log/httpd/access\_log

```
10.10.2.100 - - [22/Mar/2020:20:01:38 +0800] "GET /wp-admin/load-  
scripts.php?c=0&load%5Bchunk_0%5D=jquery-core,jquery-migrate,utils&ver=5.3.2 HTTP/1.1"  
200 108765 "http://10.10.2.105/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0)  
Gecko/20100101 Firefox/74.0"
```

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



```
10.10.2.100 - - [22/Mar/2020:20:01:38 +0800] "GET /wp-admin/load-  
scripts.php?c=0&load%5Bchunk_0%5D=jquery-core,jquery-migrate,utils&ver=5.3.2 HTTP/1.1" 200 108765  
"http://10.10.2.105/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101  
Firefox/74.0"
```

1. 10.10.2.100
2. -
3. -
4. [22/Mar/2020:20:01:38
5. +0800]
6. "GET
7. /wp-admin/load-scripts.php?c=0&load%5Bchunk\_0%5D=jquery-core,jquery-migrate,utils&ver=5.3.2
8. HTTP/1.1 "
9. 200
10. 108765
11. "http://10.10.2.105/wp-admin/ "
12. "Mozilla/5.0
13. (Windows
14. NT
15. 10.0;
16. Win64;
17. x64;
18. rv:74.0)
19. Gecko/20100101
20. Firefox/74.0"

以空格为分隔符将日志拆分为字段





操作视频 / 现场演示

- ✓ 任务7：使用CentOS命令分析Apache日志
  - 任务目标：
    - 理解Apache日志格式和字段含义
    - 使用awk、uniq、sort命令实现日志分析





任务总结 / 任务扩展

## ✓ 日志分析需求

- 访问量最高的10个来源IP地址
- 访问次数最多的文件或页面
- 访问量最高的视频文件
- 文件大小超过40M的视频文件的访问量
- 统计网站流量
- 发生404响应的用户请求
- 统计http status
- 网站访问最常用的协议
- 用户访问网站最常用的浏览器
- 访问网站的客户端设备





## 命令指南 / 操作引导

1. #访问量最高的10个来源IP地址  
2. `cat /var/log/httpd/access_log | awk '{print $1}' | sort | uniq -c | sort -nr | head -10`
3. #访问次数最多的文件或页面  
4. `cat /var/log/httpd/access_log | awk '{print $11}' | sort | uniq -c | sort -nr | head -20`
5. #访问量最高的视频文件  
6. `cat /var/log/httpd/access_log | awk '($7~/\.\mp4/){ print $7}' | sort -nr | uniq -c | head -10`
7. 3文件大小超过40M的视频文件的访问量  
8. `cat /var/log/httpd/access_log | awk '($10 > 40000000 && $7~/\.\mp4/){print $7}' | sort -n | uniq -c | sort -nr | head -100`
9. #统计网站流量  
10. `cat /var/log/httpd/access_log | awk '{sum+=$10} END {print sum/1024/1024 "MB"}`
11. #发生404响应的用户请求  
12. `cat /var/log/httpd/access_log | awk '($9~/404/)' | awk '{print $9,$7}' | sort | uniq -c`
13. #统计http status  
14. `cat /var/log/httpd/access_log | awk '{print $9}' | sort | uniq -c | sort -nr`
15. #网站访问最常用的协议  
16. `cat /var/log/httpd/access_log | awk '{print $8}' | sort | uniq -c | sort -nr`
17. #用户访问网站最常用的浏览器  
18. `cat /var/log/httpd/access_log | awk '{print $20}' | sort | uniq -c | sort -nr | head -20`
19. #访问网站的客户端设备使用情况  
20. `cat /var/log/httpd/access_log | awk '{print $16 "\t" $15 "\t" $17 "\t" $20 "\t" $19}' | sort | uniq -c | sort -nr | head -20`



## 6.Apache日志分析

### 6.3 任务8

#### 任务8：使用ApacheTop工具实时分析Apache日志

步骤1：安装ApacheTop

步骤2：明确日志分析需求

步骤3：使用apachetop工具实现apache日志的实时分析

步骤4：阅读apachetop日志分析结果





操作视频 / 现场演示

- ✓ 任务8：使用ApacheTop工具实时分析Apache日志
  - 任务目标：
    - 使用apachetop工具实现Apache日志实时分析
    - 阅读apachetop的分析结果





## 命令指南 / 操作引导

1. #安装epel扩展库
2. root@VM-Project-01-Task-02 ~]# yum install epel-release
3. #使用yum工具安装apachetop
4. [root@VM-Project-01-Task-02 ~]# yum install -y apachetop
5. #查看apachetop工具的帮助，了解其用法
6. [root@VM-Project-01-Task-02 ~]# apachetop --help
7. #使用apachetop实时分析Apache日志
8. [root@VM-Project-01-Task-02 ~]# apachetop -qI
9. #访问Apache发布的网站，并查看动态分析结果

```

Last hit: 12:44:02      atop runtime: 0 days, 00:01:10      12:44:14
All:      56 reqs ( 0.9/sec)    2375.7K ( 38.3K/sec)    42.4K/req
2xx:      56 ( 100%) 3xx:      0 ( 0.0%) 4xx:      0 ( 0.0%) 5xx:      0 ( 0.0%)
R ( 30s): 17 reqs ( 0.6/sec)    183.5K ( 6263.5B/sec)  10.8K/req
2xx:      17 ( 100%) 3xx:      0 ( 0.0%) 4xx:      0 ( 0.0%) 5xx:      0 ( 0.0%)

  REQS REQ/S    KB KB/S URL
  ---- -
    11 0.41 176.6 6.5*/server-status?refresh=1
     6 0.38  6.9 0.4 /server-status?auto
  
```







任务总结 / 任务扩展

## ✓ ApacheTop

- ApacheTop是一个命令行工具，它可通过分析Apache服务器的日志文件，向用户展示服务器的实时运行情况，并可查看访问者正在查看那些文件和访问者的IP等信息。
- ApacheTop的本质是：tail -f /var/log/httpd/access\_log
- <https://www.cae.me.uk/p/projects.apachetop>
  - ApacheTop is a Common Log Format analyser I wrote in C++ around 2003. It displays realtime statistics about webserver log entries in a curses interface, inspired by top.

## ✓ Apache日志分析软件

- Webalizer
  - The Webalizer is a fast, free web server log file analysis program. It produces highly detailed, easily configurable usage reports in HTML format, for viewing with a standard web browser.
  - <http://www.webalizer.org>
- Awstats
  - AWStats是一款功能强大的免费工具，以图形方式生成高级Web，流媒体，ftp或邮件服务器统计信息。
  - <https://awstats.sourceforge.io>





任务总结 / 任务扩展

## ✓ 大数据服务实现Apache日志分析

- 七牛云：智能日志管理平台
  - <https://www.qiniu.com/products/insight>
- 阿里云：阿里云·Elasticsearch
  - <https://data.aliyun.com/product/elasticsearch>

