

Pré-requisitos

Passo 1 — Instalando o Apache

Passo 2 — Ajustando o Firewall

Passo 3 — Verificando seu Servidor Web

Passo 4 — Configurando Hosts Virtuais (Recomendado)

Conclusão

TUTORIAL

Como Instalar o Servidor Web Apache no Ubuntu 20.04 [Quickstart]

By Erin Glass
Posted May 15, 2020 @ 800

Português

Introdução

O servidor HTTP Apache é o servidor Web mais amplamente usado no mundo. Ele fornece muitas características poderosas, incluindo módulos carregáveis ​​dinamicamente, suporte robusto de mídia e uma integração extensa com outros softwares populares.

Neste guia, vamos explicar como instalar um servidor web Apache em seu servidor Ubuntu 20.04. Para uma versão mais detalhada deste tutorial, por favor, consulte [Como Instalar o Servidor Web Apache no Ubuntu 20.04](#).

Pré-requisitos

Antes de iniciar este guia, você deve ter o seguinte:

- Um servidor Ubuntu 20.04 e um usuário regular não-root com privilégios sudo. Além disso, você precisará habilitar um firewall básico para bloquear portas não essenciais. Você pode aprender como configurar uma conta de usuário regular e configurar um firewall para seu servidor seguindo nosso guia de [Configuração inicial de servidor para o Ubuntu 20.04](#).

Quando tiver uma conta disponível, faça login com seu usuário não root para começar.

RELATED

Como instalar e configurar um servidor OpenVPN no Ubuntu 20.04

Tutorial

Como Configurar um Firewall com o UFW no Ubuntu 20.04

Tutorial

Passo 1 — Instalando o Apache

O Apache está disponível dentro dos repositórios de software padrão do Ubuntu, então você pode instalá-lo utilizando ferramentas de gerenciamento de pacotes convencionais.

Atualize seu índice de pacotes local:

\$ sudo apt update

Instale o pacote apache2:

\$ sudo apt install apache2

Passo 2 — Ajustando o Firewall

Verifique os perfis de aplicativo ufw disponíveis:

\$ sudo ufw app list

Output

Available applications:

Apache

Apache Full

Apache Secure

OpenSSH

Vamos habilitar o perfil mais restritivo de ainda permitirá o tráfego que você configurou, permitindo tráfego na porta 80 (tráfego normal, não criptografado):

\$ sudo ufw allow 'Apache'

Verifique a mudança:

\$ sudo ufw status

Output

Status: active

To Action From

--

OpenSSH ALLOW Anywhere

Apache ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

Apache (v6) ALLOW Anywhere (v6)

Passo 3 — Verificando seu Servidor Web

Verifique com o sistema init systemd para garantir que o serviço está funcionando digitando:

\$ sudo systemctl status apache2

Output

apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese-

Active: active (running) since Tue 2020-04-28 23:06:40 UTC; 56s ago

Docs: https://httpd.apache.org/docs/2.4/

Main PID: 13785 (apache2)

Tasks: 55 (Limit: 1137)

Memory: 5.3M

Group: /system.slice/apache2.service

└─13785 /usr/sbin/apache2 -k start

└─13787 /usr/sbin/apache2 -k start

└─13788 /usr/sbin/apache2 -k start

Acesse a página de destino padrão do Apache para confirmar que o software está funcionando corretamente através do seu endereço de IP:

http://your_server_ip

Você deve receber a página web padrão do Apache no Ubuntu 20.04.

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

/etc/apache2/
├-- apache2.conf
├-- ports.conf
├-- mods-enabled/
│ ├── *.conf
├-- conf-enabled/
│ ├── *.conf
├-- sites-enabled/
│ ├── *.conf

• apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

• ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

• Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

• They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **a2enmod**, **a2dismod**, **a2ensite**, **a2dissite**, **a2enconf**, **a2disconf**. See their respective man pages for detailed information.

• The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www, public_html directories (when enabled) and /usr/share (for web applications). If your site is using a web document root located elsewhere (such as in /srv) you may need to whitelist your document root directory in /etc/apache2/mods-enabled/*.conf.

The default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Passo 4 — Configurando Hosts Virtuais (Recomendado)

As usar o servidor Web Apache, você pode usar *hosts virtuais* (similares a blocos de domínio do Nginx) para encapsular detalhes de configuração e hospedar mais de um domínio de um único servidor. Vamos configurar um domínio chamado **your_domain**, mas você deve **substituí-lo por seu próprio nome de domínio**. Para aprender mais sobre como configurar um nome de domínio com a DigitalOcean, por favor, consulte nossa [Introdução ao DNS da DigitalOcean](#).

Orie o diretório para o **your_domain**:

\$ sudo mkdir /var/www/your_domain

Atribua a propriedade do diretório:

\$ sudo chown -R \$USER:\$USER /var/www/your_domain

As permissões dos seus web roots devem estar corretas se você ainda não tiver modificado o valor umask, mas você pode certificar-se digitando:

\$ sudo chmod -R 755 /var/www/your_domain

Crie uma página de amostra index.html utilizando o nano ou seu editor favorito:

\$ nano /var/www/your_domain/index.html

Dentro, adicione a seguinte amostra HTML:

/var/www/your_domain/index.html

<html>

<head>

<title>Welcome to Your_domain!</title>

</head>

<body>

<h1>Success! The your_domain virtual host is working!</h1>

</body>

</html>

Salve e feche o arquivo quando você terminar.

Crie um novo arquivo de host virtual em /etc/apache2/sites-available/your_domain.conf:

\$ sudo nano /etc/apache2/sites-available/your_domain.conf

Cole no seguinte bloco de configuração, atualizado para nosso novo diretório e nome de domínio:

/etc/apache2/sites-available/your_domain.conf

<VirtualHost *:80>

ServerAdmin webmaster@localhost

ServerName your_domain

ServerAlias your_domain

DocumentRoot /var/www/your_domain

ErrorLog \${APACHE_LOG_DIR}/error.log

CustomLog \${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

Salve e feche o arquivo quando você terminar.

Habilite o arquivo com o a2ensite:

\$ sudo a2ensite your_domain.conf

Desabilite o site padrão definido em 000-default.conf:

\$ sudo a2dissite 000-default.conf

Teste para erros de configuração:

\$ sudo apache2ctl configtest

Você deve receber a seguinte saída:

Output

Syntax OK

Reinicie o Apache para implementar as suas alterações:

\$ sudo systemctl restart apache2

O Apache agora deve estar atendendo seu nome de domínio. Teste isso navegando para http://your_domain, onde você deve ver algo assim:

Success! The your_domain virtual host is working!

Conclusão

Agora que você tem seu servidor Web instalado, você tem muitas opções para o tipo de conteúdo a oferecer e as tecnologias que quiser usar para criar uma experiência mais rica.

Se você quiser construir uma pilha de aplicação mais completa, verifique este artigo sobre [Como Configurar uma pilha LAMP no Ubuntu 20.04](#).

By Erin Glass
Posted May 15, 2020 @ 800

Como você avalia a qualidade da tradução?

Was this helpful?

Yes

No

Report an issue

Related

TUTORIAL

Como instalar e configurar um servidor OpenVPN no Ubuntu 20.04

Uma Ubuntu Private Network (VPN) permite que você atravesse redes não ...

TUTORIAL

Como Configurar um Firewall com o UFW no Ubuntu 20.04

O UFW, ou Uncomplicated Firewall, é uma interface de gerenciamento simplificada de firewall

TUTORIAL

Como Instalar a Distribuição Anaconda Python no Ubuntu 20.04

O Anaconda é um gerenciador de pacotes de código aberto, gerenciador de ambiente e ...

TUTORIAL

Como Instalar o Nginx no Ubuntu 20.04 [Quickstart]

O Nginx é um dos servidores Web mais populares no mundo e é responsável por hospedar

Still looking for an answer?

Ask a question

Search for more help

0 Comments

B I [icons]

Leave a comment...

Sign In to Comment

CC BY-NC-SA

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

COVID-19 SUPPORT PROGRAM

Working on something related to COVID-19? DigitalOcean would like to help.

Featured on Community

Kubernetes Course

Learn Python 3

Machine Learning in Python

Getting started with Go

Intro to Kubernetes

DigitalOcean Products

Droplets

Managed Databases

Managed Kubernetes

Spaces Object Storage

Marketplace

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

Learn More

Company

About

Leadership

Blog

Careers

Partners

Referral Program

Press

Legal & Security

Products

Products Overview

Pricing

Droplets

Kubernetes

Managed Databases

Spaces

Marketplace

Load Balancers

Block Storage

Tools & Integrations

API

Documentation

Release Notes

Community

Tutorials

Q&A

Tools and Integrations

Tags

Product Ideas

Meetups

Write for DOnations

Droplets for Demos

Hatch Startup Program

Shop Swagger

Research Program

Open Source

Code of Conduct

Contact

Get Support

Trouble Signing In?

Sales

Report Abuse

System Status

Sign up for our newsletter.

Get the latest tutorials on SysAdmin and open source topics.

Enter your email

Sign

© 2020 DigitalOcean, LLC. All rights reserved.

SCROLL TO TOP