



CONTENTS

Introdução

Pré-requisitos

Passo 1 — Instalando o Apache

Passo 2 — Ajustando o Firewall

Passo 3 — Verificando seu Servidor Web

Passo 4 — Gerenciando o Processo Apache

Passo 5 — Configurando Hosts Virtuais (Recomendado)

Passo 6 — Familiarizando-se com Arquivos e Diretórios Importantes do Apache

Conteúdo

Configuração do Servidor

Registros do Servidor

Conclusão

RELATED

Como instalar o PostgreSQL no Ubuntu 20.04: [Guia de início rápido]

[View](#) [↗](#)

Como Instalar a Distribuição Anaconda Python no Ubuntu 20.04

[View](#) [↗](#)

// Tutorial //

Como Instalar o Servidor Web Apache no Ubuntu 20.04

Published on May 15, 2020

[Apache](#) [Ubuntu](#) [Ubuntu 20.04](#)



By [Erin Glass](#)

Senior Developer Educator

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.

Pré-requisitos

Antes de iniciar este guia, você deve ter um usuário regular e não-root com privilégios sudo configurado no seu servidor. 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.

Passo 1 – Instalando o Apache

O Apache está disponível dentro dos repositórios padrão de software do Ubuntu, possibilitando que ele seja instalado utilizando ferramentas convencionais de gerenciamento de pacotes.

Vamos começar atualizando o índice de pacotes local para refletir as últimas alterações do upstream:

```
$ sudo apt update
```

Copy

Então, instale o pacote `apache2`:

```
$ sudo apt install apache2
```

Copy

Após confirmar a instalação, o `apt` irá instalar o Apache e todas as dependências necessárias.

Passo 2 – Ajustando o Firewall

Antes de testar o Apache, é necessário modificar as configurações do firewall para permitir o acesso externo às portas Web padrão. Supondo que você tenha seguido as instruções nos pré-requisitos, você deve ter um firewall UFW configurado para restringir o acesso ao seu servidor.

Durante a instalação, o Apache registra-se com o UFW para fornecer alguns perfis de aplicativo que podem ser usados para habilitar ou desabilitar o acesso ao Apache através do firewall.

Liste os perfis de aplicativo `ufw` digitando:

```
$ sudo ufw app list
```

Copy

Você receberá uma lista dos perfis de aplicações:

Output

```
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

Como indicado pela saída, há três perfis disponíveis para o Apache:

- **Apache:** Este perfil abre apenas a porta 80 (normal, tráfego Web não criptografado)
- **Apache Full:** Este perfil abre ambas as portas 80 (normal, tráfego Web não criptografado) e 443 (tráfego TLS/SSL criptografado)
- **Apache Secure:** Este perfil abre apenas a porta 443 (tráfego TLS/SSL criptografado)

É recomendável que habilite o perfil mais restritivo que ainda assim permitirá o tráfego que você configurou. Como ainda não configurou o SSL para seu servidor neste guia, nós precisaremos apenas permitir o tráfego na porta 80:

```
$ sudo ufw allow 'Apache' Copy
```

Você pode verificar a mudança digitando:

```
$ sudo ufw status Copy
```

A saída fornecerá uma lista de tráfego HTTP permitido:

Output

```
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

Como indicado pela saída, o perfil foi ativado para permitir o acesso ao servidor web Apache.

Passo 3 – Verificando seu Servidor Web

No final do processo de instalação, o Ubuntu 20.04 inicia o Apache. O servidor Web já deve estar em funcionamento.

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

```
$ sudo systemctl status apache2 Copy
```

Output

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 29435 (apache2)
    Tasks: 55 (limit: 1137)
   Memory: 8.0M
    CGroup: /system.slice/apache2.service
            └─29435 /usr/sbin/apache2 -k start
            └─29437 /usr/sbin/apache2 -k start
            └─29438 /usr/sbin/apache2 -k start
```



Como confirmado por esta saída, o serviço foi iniciado com sucesso. No entanto, a melhor maneira de testar isso é solicitando uma página do Apache.

Você pode acessar a página de destino padrão do Apache para confirmar que o software está funcionando corretamente através do seu endereço IP: Se você não sabe o endereço IP do seu servidor, você pode obtê-lo de algumas maneiras diferentes a partir da linha de comando.

Tente digitar isso no prompt de comando do seu servidor:

```
$ hostname -I
```

Copy

Você receberá alguns endereços separados por espaços. Você pode testar cada um no seu navegador web para ver se eles funcionam.

Outra opção é usar a ferramenta Icanhazip, que deve dar a você o endereço IP público como visto de outro local na Internet:

```
$ curl -4 icanhazip.com
```

Copy

Quando você tiver o endereço IP do seu servidor, digite-o na barra de endereço do seu navegador:

```
http://your_server_ip
```

Você deve ver a página web padrão do Apache no Ubuntu 20.04:





ubuntu

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
|   |-- *.load
|   |-- *.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`, and `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/apache2.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.

Esta página indica que o Apache está funcionando corretamente. Ela também inclui algumas informações básicas sobre arquivos importantes do Apache e localizações de diretórios importantes.

Passo 4 – Gerenciando o Processo Apache

Agora que você tem seu servidor web funcionando, vamos ver alguns comandos básicos de gerenciamento usando o `systemctl`

Para parar seu servidor Web, digite:

```
$ sudo systemctl stop apache2
```

Copy

Para iniciar o servidor quando ele for parado, digite:

```
$ sudo systemctl start apache2
```

Copy



Para parar e então iniciar o serviço novamente, digite:

```
$ sudo systemctl restart apache2
```

Copy

Se você estiver simplesmente fazendo alterações de configuração, o Apache geralmente pode recarregar sem quedas na conexão. Para fazer isso, utilize este comando:

```
$ sudo systemctl reload apache2
```

Copy

Por padrão, o Apache está configurado para iniciar automaticamente quando o servidor for iniciado. Se isso não é o que você quer, desative este comportamento digitando:

```
$ sudo systemctl disable apache2
```

Copy

Para reativar o serviço de inicialização no boot, digite:

```
$ sudo systemctl enable apache2
```

Copy

O Apache deve iniciar automaticamente quando o servidor for iniciado outra vez.

Passo 5 – Configurando Hosts Virtuais (Recomendado)

Ao usar o servidor Web Apache, você pode usar *hosts virtuais* (similares a blocos de servidor no 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**. Se você estiver configurando um nome de domínio com a DigitalOcean, consulte nossa [Documentação de Rede](#).

O Apache no Ubuntu 20.04 tem um bloco de servidor habilitado por padrão que está configurado para servir documentos do diretório `/var/www/html`. Enquanto isso funciona bem para um único site, ele pode tornar-se indevido se você estiver hospedando vários sites. Ao invés de modificar o `/var/www/html`, vamos criar uma estrutura de diretórios dentro do `/var/www` para um site **your_domain**, deixando o `/var/www/html` no lugar como o diretório padrão para ser exibido se um pedido de cliente não corresponder a nenhum outro site.

Crie o diretório para o **your_domain** como segue:

```
$ sudo mkdir /var/www/your_domain
```

Copy

Em seguida, atribua a propriedade do diretório com a variável de ambiente `$USER`:

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

Copy

As permissões dos seus web roots devem estar corretas se você não tiver modificado seu valor de umask, que define permissões padrão de arquivos. Para garantir que suas permissões estejam corretas e permitam que o proprietário leia, escreva e execute os arquivos, enquanto concede apenas permissões de leitura e execução para grupos e outros, você pode digitar o seguinte comando:

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

Copy

A seguir, crie uma página de amostra `index.html` utilizando o `nano` ou seu editor favorito:

```
$ sudo nano /var/www/your_domain/index.html
```

Copy

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>
```

Copy




```
</body>
</html>
```

Salve e feche o arquivo quando você terminar.

Para que o Apache sirva este conteúdo, é necessário criar um arquivo de host virtual com as diretivas corretas. Em vez de modificar o arquivo de configuração padrão localizado em `/etc/apache2/sites-available/000-default.conf` diretamente, vamos criar um novo em `/etc/apache2/sites-available/your_domain.conf`:

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

 Copy

Cole no seguinte bloco de configuração, que é similar ao padrão, mas 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 www.your_domain
    DocumentRoot /var/www/your_domain
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

 Copy

Note que atualizamos o `DocumentRoot` para nosso novo diretório e o `ServerAdmin` para um e-mail que o administrador do site **your_domain** pode acessar. Também adicionamos duas diretivas: o `ServerName`, que estabelece o domínio base que deve corresponder a essa definição de host virtual, e o `ServerAlias`, que define nomes adicionais que devem corresponder como se fossem o nome base.

Salve e feche o arquivo quando você terminar.

Vamos habilitar o arquivo com a ferramenta `a2ensite`:

```
$ sudo a2ensite your_domain.conf
```

 Copy

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

```
$ sudo a2dissite 000-default.conf
```

 Copy

Em seguida, vamos testar à procura de erros de configuração:

```
$ sudo apache2ctl configtest
```

 Copy

Você deve receber a seguinte saída:

```
Output
Syntax OK
```

Reinicie o Apache para implementar as suas alterações:

```
$ sudo systemctl restart apache2
```

 Copy

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

Success! The your_domain virtual host is working!



Passo 6 – Familiarizando-se com Arquivos e Diretórios Importantes do Apache

Agora que você sabe como gerenciar o serviço do Apache, você deve gastar alguns minutos para familiarizar-se com alguns diretórios e arquivos importantes.

Conteúdo

- `/var/www/html` : O conteúdo Web em si, que por padrão apenas consiste na página Apache padrão que você viu antes, é servido do diretório `/var/www/html`. Isso pode ser alterado mudando os arquivos de configuração do Apache.

Configuração do Servidor

- `/etc/apache2` : o diretório de configuração do Apache. Todos os arquivos de configuração do Apache residem aqui.
- `/etc/apache2/apache2.conf` : O arquivo de configuração principal do Apache. Isso pode ser modificado para fazer alterações na configuração global do Apache. Este arquivo é o responsável por carregar muitos dos outros arquivos no diretório de configuração.
- `/etc/apache2/ports.conf` : Este arquivo especifica as portas nas quais o Apache irá escutar. Por padrão, o Apache escuta na porta 80 e adicionalmente escuta na porta 443 quando um módulo que fornece capacidades SSL está ativo.
- `/etc/apache2/sites-available/` : O diretório onde hosts virtuais de cada site podem ser armazenados. O Apache não usará os arquivos de configuração encontrados neste diretório a menos que estejam ligados ao diretório `sites-enabled`. Normalmente, todas as configurações de bloco do servidor são feitas neste diretório e então habilitadas ligando-as ao outro diretório com o comando `a2ensite`.
- `/etc/apache2/sites-enabled/` : O diretório onde hosts virtuais habilitados de cada site são armazenados. Normalmente, eles são criados ligando arquivos de configuração encontrados no diretório `sites-available` com o `a2ensite`. O Apache lê os arquivos de configuração e links encontrados neste diretório quando inicia ou recarrega para compilar uma configuração completa.
- `/etc/apache2/conf-available/`, `/etc/apache2/conf-enabled/` : Estes diretórios têm a mesma relação que os diretórios `sites-available` e `sites-enabled`, mas são usados para armazenar fragmentos de configuração que não pertencem em um host virtual. Arquivos no diretório `conf-available` podem ser habilitados com o comando `a2enconf` e desabilitados com o comando `a2disconf`.
- `/etc/apache2/mods-available/`, `/etc/apache2/mod-enabled/` : Estes diretórios contêm os módulos disponíveis e habilitados, respectivamente. Arquivos com final `.load` contêm fragmentos para carregar módulos específicos, enquanto os arquivos com final `.conf` contêm a configuração para esses módulos. Módulos podem ser habilitados e desabilitados utilizando os comandos `a2enmod` e `a2dismod`.

Registros do Servidor

- `/var/log/apache2/access.log` : Por padrão, cada solicitação feita para seu servidor é gravada neste arquivo de registro a menos que o Apache esteja configurado para fazer de outro modo.
- `/var/log/apache2/error.log` : Por padrão, todos os erros são gravados neste arquivo. A diretiva `LogLevel` na configuração do Apache especifica quanto detalhe os registros de erros irão conter.

Conclusão

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

Se você quiser construir uma pilha de aplicação mais completa, você pode ler este artigo sobre [como configurar uma pilha LAMP no Ubuntu 20.04](#).

Want to learn more? Join the DigitalOcean Community!

Join our DigitalOcean community of over a million developers for free! Get help and share knowledge in our Questions & Answers section, find tutorials and tools that will help you grow as a developer and scale your project or business, and subscribe to topics of interest.

Sign up →

