



CONTENTS

Introdução

Pré-requisitos

Passo 1 - Instalando o Nginx

Passo 2 — Como ajustar o Firewall

Passo 3 — Verificando seu Servidor Web

Passo 4 — Gerenciando o processo do Nginx

Passo 5 — Configurando Blocos do Servidor (Recomendado)

Passo 6 — Familiarizando-se com arquivos e diretórios importantes do Nginx

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 Nginx no Ubuntu 20.04

Published on May 15, 2020

Ubuntu Nginx Ubuntu 20.04



By [Erin Glass](#)

Senior Developer Educator

Português





Introdução

O [Nginx](#) é um dos servidores Web mais populares no mundo e é responsável por hospedar alguns dos sites de maior tráfego na Internet. Ele é uma escolha leve que pode ser usado como servidor web ou proxy reverso.

Neste guia, vamos discutir como instalar o Nginx em seu servidor Ubuntu 20.04, ajustar o firewall, gerenciar o processo do Nginx e configurar os blocos de servidor para hospedar mais de um domínio em um único servidor.

Pré-requisitos

Antes de iniciar este guia, você deve ter um usuário regular e não-root com privilégios sudo configurado no seu servidor. Você pode aprender como configurar uma conta de usuário regular 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 Nginx

Como o Nginx está disponível nos repositórios padrão do Ubuntu, é possível instalá-lo a partir desses repositórios usando o sistema de pacotes do `apt`.

Como essa é nossa primeira interação com o sistema de pacotes do `apt` nesta sessão, também vamos atualizar nosso índice de pacotes local para que tenhamos acesso às listagens de pacotes mais recentes. Depois disso, podemos instalar o `nginx`:

```
$ sudo apt update
$ sudo apt install nginx
```

Copy

Depois de aceitar o procedimento, o `apt` instalará o Nginx e quaisquer dependências necessárias no seu servidor.

Passo 2 – Como ajustar o Firewall

Antes de testar o Nginx, o software de firewall precisa ser ajustado para permitir o acesso ao serviço. O Nginx registra-se como um serviço com o `ufw` após a instalação, tornando-simples permitir o acesso ao Nginx

Liste as configurações do aplicativo com as quais o `ufw` sabe trabalhar digitando:

```
$ sudo ufw app list
```

Copy



Você deve obter uma lista dos perfis dos aplicativos:

Output

```
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
```

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

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

É recomendável que você habilite o perfil mais restritivo que ainda assim permitirá o tráfego que você configurou. Agora, precisaremos apenas permitir o tráfego na porta 80.

Permita isso digitando:

```
$ sudo ufw allow 'Nginx HTTP'
```

 Copy

Você pode verificar a mudança digitando:

```
$ sudo ufw status
```

 Copy

A saída indicará qual tráfego HTTP é permitido:

Output

```
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Nginx HTTP	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Nginx HTTP (v6)	ALLOW	Anywhere (v6)

Passo 3 – Verificando seu Servidor Web

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

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

```
$ systemctl status nginx
```

 Copy

Output

```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-04-20 16:08:19 UTC; 3 days ago
     Docs: man:nginx(8)
  Main PID: 2369 (nginx)
    Tasks: 2 (limit: 1153)
   Memory: 3.5M
    CGroup: /system.slice/nginx.service
            └─2369 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─2380 nginx: worker process
```

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



Você pode acessar a página de inicial padrão do Apache para confirmar que o software está funcionando corretamente navegando para o endereço IP do seu servidor: Se você não sabe o endereço IP do seu servidor, você pode encontrá-lo usando a ferramenta icanhazip.com, que lhe dará o endereço IP público conforme recebido a partir 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 receber a página inicial padrão do Nginx:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Se você estiver nesta página, seu servidor está funcionando corretamente e está pronto para ser gerenciado.

Passo 4 – Gerenciando o processo do Nginx

Agora que você tem seu servidor Web em funcionamento, vamos rever alguns comandos básicos de gerenciamento.

Para parar seu servidor Web, digite:

```
$ sudo systemctl stop nginx
```

Copy

Para iniciar o servidor quando ele estiver parado, digite:

```
$ sudo systemctl start nginx
```

Copy

Para parar e então iniciar o serviço novamente, digite:

```
$ sudo systemctl restart nginx
```

Copy

Se você estiver simplesmente fazendo alterações de configuração, o Nginx geralmente pode recarregar sem quedas na conexão. Para fazer isso, digite:

```
$ sudo systemctl reload nginx
```

Copy

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

```
$ sudo systemctl disable nginx
```

Copy

Para reativar o serviço para iniciar no boot, digite:

```
$ sudo systemctl enable nginx
```

Copy

Agora, você aprendeu os comandos básicos de gerenciamento e deve estar pronto para configurar o site para hospedar mais de um domínio.

Passo 5 – Configurando Blocos do Servidor (Recomendado)



Ao usar o servidor Web Nginx, os *server blocks* (similares aos hosts virtuais no Apache) podem ser usados 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](#).

O Nginx no Ubuntu 20.04 tem um bloco de servidor habilitado por padrão que está configurado para exibir 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. Em vez de modificar o `/var/www/html`, vamos criar uma estrutura de diretórios dentro do `/var/www` para nosso site **your_domain**, deixando o `/var/www/html` intacto como o diretório padrão a ser servido se um pedido de cliente não corresponder a nenhum outro site.

Crie o diretório para o **your_domain** da seguinte forma, utilizando o sinalizador `-p` para criar quaisquer diretórios pai necessários:

```
$ sudo mkdir -p /var/www/your_domain/html
```

Copy

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

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

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:

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

Copy

Dentro, adicione a seguinte amostra HTML:

```
                                /var/www/your_domain/html/index.html

<html>
  <head>
    <title>Welcome to your_domain!</title>
  </head>
  <body>
    <h1>Success! The your_domain server block is working!</h1>
  </body>
</html>
```

Copy

Salve e feche o arquivo digitando `CTRL` e `X`, depois `Y` e `ENTER` quando você terminar.

Para que o Nginx exiba este conteúdo, é necessário criar um bloco de servidor com as diretivas corretas. Em vez de modificar o arquivo de configuração padrão diretamente, vamos fazer um novo em `/etc/nginx/sites-available/example.com`:

```
$ sudo nano /etc/nginx/sites-available/your_domain
```

Copy

Cole no seguinte bloco de configuração, que é similar ao padrão, mas atualizado para nosso novo diretório e nome de domínio:

```
                                /etc/nginx/sites-available/your_domain

server {
  listen 80;
  listen [::]:80;

  root /var/www/your_domain/html;
  index index.html index.htm index.nginx-debian.html;

  server_name your_domain www.your_domain;

  location / {
```

Copy

```

        try_files $uri $uri/ =404;
    }
}

```

Note que atualizamos a configuração do `root` para nosso novo diretório e o `server_name` para nosso nome de domínio.

Em seguida, vamos habilitar o arquivo criando um link dele para o diretório `sites-enabled`, de onde o Nginx lê durante a inicialização:

```
$ sudo ln -s /etc/nginx/sites-available/your_domain /etc/nginx/sites-enabled/
```

Copy

Agora, dois blocos de servidor estão habilitados e configurados para responder às solicitações baseados em suas diretivas `listen` e `server_name` (você pode ler mais sobre como o Nginx processa essas diretivas [aqui](#)):

- `your_domain`: irá responder às solicitações para `your_domain` e `www.your_domain`.
- `default`: responderá a quaisquer pedidos na porta 80 que não correspondam aos outros dois blocos.

Para evitar um possível problema de memória de hash que possa surgir ao adicionar nomes adicionais de servidor, é necessário ajustar um valor único no arquivo `/etc/nginx/nginx.conf`. Abra o arquivo:

```
$ sudo nano /etc/nginx/nginx.conf
```

Copy

Encontre a diretiva `server_names_hash_bucket_size` e remova o símbolo `#` para descomentar a linha: Se você estiver usando o `nano`, você pode procurar rapidamente por palavras no arquivo pressionando `CTRL` e `w`.

```

/etc/nginx/nginx.conf

...
http {
    ...
    server_names_hash_bucket_size 64;
    ...
}
...

```

Salve e feche o arquivo quando você terminar.

Em seguida, teste para garantir que não haja erros de sintaxe em qualquer um dos seus arquivos do Nginx:

```
$ sudo nginx -t
```

Copy

Se não houver problemas, reinicie o Nginx para habilitar suas alterações:

```
$ sudo systemctl restart nginx
```

Copy

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

Success! The example.com server block is working!

Passo 6 – Familiarizando-se com arquivos e diretórios importantes do Nginx

Agora que sabe como gerenciar o serviço do Nginx, 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 Nginx padrão que você viu antes, é servido fora do diretório `/var/www/html` . Isso pode ser alterado mudando os arquivos de configuração do Nginx.

Configuração do Servidor

- `/etc/nginx` : o diretório de configuração do Nginx. Todos os arquivos de configuração do Nginx residem aqui.
- `/etc/nginx/nginx.conf` : o arquivo de configuração principal do Nginx. Isso pode ser modificado para fazer alterações na configuração global do Nginx.
- `/etc/nginx/sites-available/` : o diretório onde os blocos de servidor de cada site podem ser armazenados. O Nginx não usará os arquivos de configuração encontrados neste diretório a menos que estejam ligados ao diretório `sites-enabled` . Normalmente, todas as configurações de blocos de servidor são feitas neste diretório e então habilitadas pela ligação a outro diretório.
- `/etc/nginx/sites-enabled/` : o diretório onde os blocos de servidor de cada site habilitados são armazenados. Normalmente, eles são criados pela ligação aos arquivos de configuração encontrados no diretório `sites-available` .
- `/etc/nginx/snippets` : este diretório contém fragmentos de configuração que podem ser incluídos em outro lugar na configuração do Nginx. Os segmentos de configuração potencialmente repetíveis são bons candidatos à refatoração em snippets.

Registros do Servidor

- `/var/log/nginx/access.log` : cada pedido ao seu servidor Web é registrado neste arquivo de registro a menos que o Nginx esteja configurado para fazer de outra maneira.
- `/var/log/nginx/error.log` : qualquer erro do Nginx será gravado neste registro.

Conclusão

Agora que 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 aplicações mais completa, verifique o artigo [Como Instalar Linux, Nginx, MySQL, PHP \(pilha LEMP\) 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 →

About the authors



[Erin Glass](#) Author

Senior Developer Educator

Open source advocate and lover of education, culture, and community.

