

Como Instalar Linux, Nginx, MySQL, PHP (Pilha LEMP) no Ubuntu 16.04



Posted December 23, 2016  44.5k LEMP MYSQL PHP NGINX UBUNTU UBUNTU 16.04

By: Justin Ellingwood

Introdução

A pilha de software LEMP é um grupo de softwares que pode ser utilizado para servir páginas web dinâmicas e aplicações web. Esse é um acrônimo que descreve um sistema operacional Linux, com um servidor web Nginx. Os dados de retaguarda ou backend são armazenados em um banco de dados MySQL e o processamento dinâmico é tratado pelo PHP.

Neste guia, vamos demonstrar como instalar uma pilha LEMP em um servidor Ubuntu 16.04. O sistema operacional Ubuntu cuida do primeiro requisito. Vamos descrever como colocar o restante dos componentes em funcionamento.

Pré-requisitos

Antes de completar esse tutorial, você deve ter uma conta regular, que não seja root, com privilégios `sudo`, configurada em seu servidor. Você pode aprender como configurar esse tipo de conta completando os passos 1-4 na [configuração inicial do servidor para Ubuntu 16.04](#).

Após ter seu usuário disponível, acesse seu servidor com esse nome de usuário. Você está agora pronto para iniciar os passos descritos neste guia.

Passo 1: Instalar o Servidor Web Nginx

A fim de exibir páginas da web para os visitantes do nosso site, vamos empregar o Nginx, um moderno e eficiente servidor web.

Todos os softwares que estaremos utilizando para esse procedimento virão diretamente dos repositórios padrão do Ubuntu. Isso significa que podemos utilizar a conjunto de gerenciamento de pacotes `apt` para completar a instalação.

Como essa é nossa primeira vez utilizando o `apt` para essa seção, devemos iniciar atualizando nosso índice local de pacotes. Podemos a seguir, instalar o servidor:

```
$ sudo apt-get update
$ sudo apt-get install nginx
```

No Ubuntu 16.04, o Nginx está configurado para iniciar a execução após a instalação.

Se você tiver o firewall `ufw` executando, como descrito no nosso guia de configuração inicial, precisaremos permitir conexões ao Nginx. O Nginx registra-se com o `ufw` após a instalação, dessa forma o procedimento é bastante simples.

É recomendável que você habilite o perfil mais restritivo que ainda permita o tráfego que você quer. Como ainda não configuramos o SSL para o nosso servidor, nesse guia, só precisaremos permitir o tráfego na porta 80.

Você pode habilitar isso digitando:

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

Você pode verificar a alteração digitando:

```
$ sudo ufw status
```

Você deve ver o tráfego HTTP permitido na saída exibida:

Output

Status: active

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

Com a nova regra de firewall adicionada, você pode testar se o servidor está funcionando ao acessar o nome de domínio de seu servidor ou seu endereço IP público em seu navegador web.

Se você não tiver um nome de domínio apontado para seu servidor e você não sabe o endereço IP público dele, você pode encontrá-lo digitando um dos seguintes comandos em seu terminal:

```
$ ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\\./.$//'
```

Esse comando irá imprimir alguns endereços IP. Você pode tentar cada um deles em seu navegador web.

Como uma alternativa, você pode verificar quais endereços IP estão acessíveis a partir de outros locais na Internet:

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

Digite um dos endereços que você recebeu em seu navegador. Ele deve levá-lo até a página inicial padrão do Nginx:

```
http://domínio_do_servidor_ou_IP
```



Se você ver a página acima, você instalou o Nginx com sucesso.

Passo 2: Instalar o MySQL para Gerenciar os Dados do Site

Agora que temos um servidor web, precisamos instalar o MySQL, um sistema de gerenciamento de banco de dados, para armazenar os dados do seu site.

Você pode instalar isso facilmente digitando:

```
$ sudo apt-get install mysql-server
```

Você será solicitado a fornecer a senha root (administrativa) para utilizar no sistema MySQL.

O software de banco de dados MySQL está instalado agora, mas a configuração ainda não está completa.

Para proteger a instalação, podemos executar um script de segurança simples que irá perguntar se queremos modificar alguns padrões inseguros. Inicie o script digitando:

```
$ sudo mysql_secure_installation
```

Você será solicitado a digitar a senha que você configurou para a conta root do MySQL. Depois, você será perguntado se você quer configurar o `VALIDATE PASSWORD PLUGIN`.

Atenção: A habilitação dessa funcionalidade é algo que deve ser avaliado. Se habilitado, senhas que não seguem o critério especificado serão rejeitadas pelo MySQL com um erro. Isso irá causar problemas se você utilizar uma senha fraca juntamente com software que configura automaticamente as credenciais de usuário do MySQL, tais como os pacotes do Ubuntu para o phpMyAdmin. É seguro deixar a validação desativada, mas você deve sempre utilizar senhas fortes e exclusivas para as credenciais do banco de dados.

Responda **y** para Sim, ou qualquer outra coisa para continuar sem a habilitação.

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
```

Press **y|Y** for Yes, any other key for No:

Você será solicitado a selecionar um nível de validação de senha. Tenha em mente que se você digitar **2**, para o nível mais forte, você receberá erros quando tentar configurar qualquer senha que não contenha números, letras maiúsculas e minúsculas, e caracteres especiais, ou que seja baseada em palavras comuns do dicionário.

There are three levels of password validation policy:

```
LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file
```

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: **1**

Se você habilitou a validação de senha, será mostrado a força da senha de root atual, e será perguntado se você quer alterar aquela senha. Se você estiver satisfeito com sua senha atual, digite **n** para "não" no prompt:

Using existing password for root.

```
Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```

Para o restante das perguntas, você deve pressionar **Y** e apertar a tecla **Enter** para cada prompt. Isso irá remover alguns usuários anônimos e o banco de dados de teste, desabilitar logins remotos de root, e carregar essas novas regras de forma que o MySQL respeite imediatamente as alterações que fizemos.

Nesse ponto, seu sistema de banco de dados está agora configurado e podemos seguir e frente.

Step 3: Instalar o PHP para Processamento

Temos agora o Nginx instalado para servir nossas páginas e o MySQL instalado para armazenar e gerenciar nossos dados. Contudo, nós ainda não temos nada que possa gerar conteúdo dinâmico. Podemos utilizar o PHP para isso.

Como o Nginx não contém processamento PHP nativo como alguns outros servidores web, precisaremos instalar o `php-fpm`, que significa "fastCGI process manager" ou "gerenciador de processos fastCGI". Vamos dizer ao Nginx para passar pedidos PHP para este software para processamento.

Podemos instalar esse módulo e pegaremos também um pacote auxiliar que permitirá ao PHP comunicar com nosso banco de dados de retaguarda. A instalação irá buscar os arquivos necessários para o núcleo do PHP. Faça isso digitando:

```
$ sudo apt-get install php-fpm php-mysql
```

Configurar o Processador PHP

Agora temos os componentes do PHP instalados, mas precisamos fazer uma pequena alteração na configuração para tornar nossa instalação mais segura.

Abra o arquivo principal de configuração do `php-fpm` com privilégios de root:

```
$ sudo nano /etc/php/7.0/fpm/php.ini
```

O que procuraremos nesse arquivo é o parâmetro que configura o `cgi.fix_pathinfo`. Ele estará comentado com um ponto e vírgula (;) e definido como "1" por padrão.

Essa é uma configuração extremamente insegura porque ela diz ao PHP para tentar executar o arquivo mais próximo se o arquivo PHP requisitado não puder ser encontrado. Isso basicamente permitiria que os usuários criassem requisições PHP de maneira que lhes permitissem executar scripts que eles não deveriam ser autorizados a executar.

Vamos alterar ambas as condições descomentando a linha e definindo-a para "0" dessa forma:

```
;/etc/php/7.0/fpm/php.ini  
  
cgi.fix_pathinfo=0
```

Salve e feche o arquivo quando tiver terminado.

Agora, precisamos apenas reiniciar nosso processador PHP digitando:

```
$ sudo systemctl restart php7.0-fpm
```

Isso irá implementar a alteração que fizemos.

Passo 4: Configurar o Nginx para Usar o Processador PHP

Agora, temos todos os componentes necessários instalados. A única alteração de configuração que ainda precisamos fazer é dizer para o Nginx para utilizar o nosso processador PHP para conteúdo dinâmico.

Fazemos isso no nível de bloco do servidor (blocos do servidor são similares ao virtual hosts do Apache). Abra o arquivo padrão de configuração de bloco do servidor Nginx digitando:

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

Atualmente, com os comentários removidos, o arquivo de bloco padrão do servidor Nginx se parece com isso:

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

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

    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Precisamos fazer algumas alterações nesse arquivo para o nosso site.

- Primeiro, precisamos adicionar `index.php` como primeiro valor de nossa diretiva `index` de forma que arquivos com nome `index.php` sejam servidos, se disponíveis, quando um diretório é requisitado.
- Podemos modificar a diretiva `server_name` para apontar para o nome de domínio do nosso servidor ou seu endereço IP público.
- Para o processamento PHP real, precisamos apenas descomentar um segmento de arquivo que trata as requisições PHP. Esse será o bloco de localização `location ~\\.php$`, o fragmento incluído `fastcgi-php.conf`, e o soquete associado com o `php-fpm`.
- Vamos descomentar também o bloco de localização que trata de arquivos `.htaccess`. O Nginx não processa esses arquivos. Se acontecer de um desses arquivos ser encontrado dentro de document root, eles não devem ser servidos aos visitantes.

As alterações que precisamos fazer estão em vermelho no texto abaixo:

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

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

    server_name domínio_do_servidor_ou_IP;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Quando você tiver feito as alterações acima, salve e feche o arquivo.

Teste seu arquivo de configuração para erros de sintaxe digitando:

```
$ sudo nginx -t
```

Se quaisquer erros forem reportados, retorne e cheque novamente seu arquivo antes de continuar.

Quando estiver pronto, recarregue o Nginx para fazer as alterações necessárias:

```
$ sudo systemctl reload nginx
```

Passo 5: Criar um Arquivo PHP para Testar a Configuração

Sua pilha LEMP deve estar agora completamente configurada. Podemos testá-la para validar que o Nginx pode manipular arquivos `.php` pelo nosso processador PHP.

Podemos fazer isso através da criação de um arquivo PHP de teste em nosso document root. Abra um novo arquivo chamado `info.php` dentro de seu document root em seu editor de textos:

```
$ sudo nano /var/www/html/info.php
```

Digite ou cole as seguintes linhas no novo arquivo. Esse é um código PHP válido que irá retornar informações sobre o seu servidor:

```
                /var/www/html/info.php


$ <?php
$ phpinfo();
```

Quando tiver finalizado, salve e feche o arquivo.

Agora, você pode visitar essa página em seu navegador web acessando o nome de domínio ou o endereço IP público do seu servidor, seguido por `/info.php` :


```
http://domínio_do_servidor_ou_IP/info.php
```

Você deve ver uma página web que foi gerada pelo PHP com informações sobre o seu servidor:

PHP Version 7.0.4-7ubuntu2

System	Linux lemp 4.4.0-18-generic #34-Ubuntu SMP Wed Apr 6 14:01:02 UTC 2016 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.0/fpm/conf.d/10-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-ctype.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-json.ini, /etc/php/7.0/fpm/conf.d/20-mysqli.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.0/fpm/conf.d/20-sysvsem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies



Se você visualizar uma página que se pareça com essa, você configurou o processamento PHP no Nginx com sucesso.

Depois de verificar que o Nginx renderiza a página corretamente, é melhor remover o arquivo que você criou, visto que ele na verdade fornece algumas dicas sobre sua configuração a usuários não autorizados, o que pode ajudá-los a entrar. Você sempre pode regerar esse arquivo se precisar posteriormente.

Por agora, remova o arquivo digitando:

```
$ sudo rm /var/www/html/info.php
```

Conclusão

Agora você deve ter uma pilha LEMP configurada em seu servidor Ubuntu 16.04. Isso lhe fornece uma fundação muito flexível para servir conteúdo web para seus visitantes.

By: Justin Ellingwood

Upvote (10) Subscribe Share



Translation:
Fernando Pimenta



We just made it easier for you to deploy faster.

TRY FREE

Related Tutorials

- How to Deploy a Symfony 4 Application to Production with LEMP on Ubuntu 18.04
- How To Install WordPress with LEMP on Debian 9

5 Comments

Leave a comment...

Log In to Comment

^ [8579](#) *March 7, 2017*



1 Obrigado pelo post, depois de dois dias procurando finalmente consegui instalar um servidor no Linux Mint.

^ [telmosports](#) *April 16, 2017*



1 Eu não entendo de programação web, nada mesmo; porém segui seu tuto para levantar um server web e deu tudo certo !!!! agora vou pra segunda parte do firewall que estou estudo acesso externo com redirecionamento ao web server e afins....mas vou deixá-lo para futuramente instalar o Wordpress ! pois quero aprender como funciona apesar de minha area é firewall e redes :)
Obrigado mesmo!

^ [gestorcmmmbca7e2824e1fa4eec](#) *April 28, 2018*



1 I do not understand much about web programming, but following the tutorial I was able to install the server.
My thanks to Justin Ellingwood for the tutorial and Fernando Pimenta.

^ [vitordesousa](#) *August 5, 2018*



1 Obrigado!



Ótimo tutorial, ajudou muito.

Eu precisei de digitar o seguinte comando para liberar o tráfego HTTP de saída.

```
$ sudo ufw enable
```



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



Copyright © 2019 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) 

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)