

Articles in this section

How to Deploy a Python App on Debian 10

How to Deploy a React.js App with Nginx on Ubuntu 20.04

How to Deploy Laravel 8 with Apache and Let's Encrypt SSL on Ubuntu 20.04

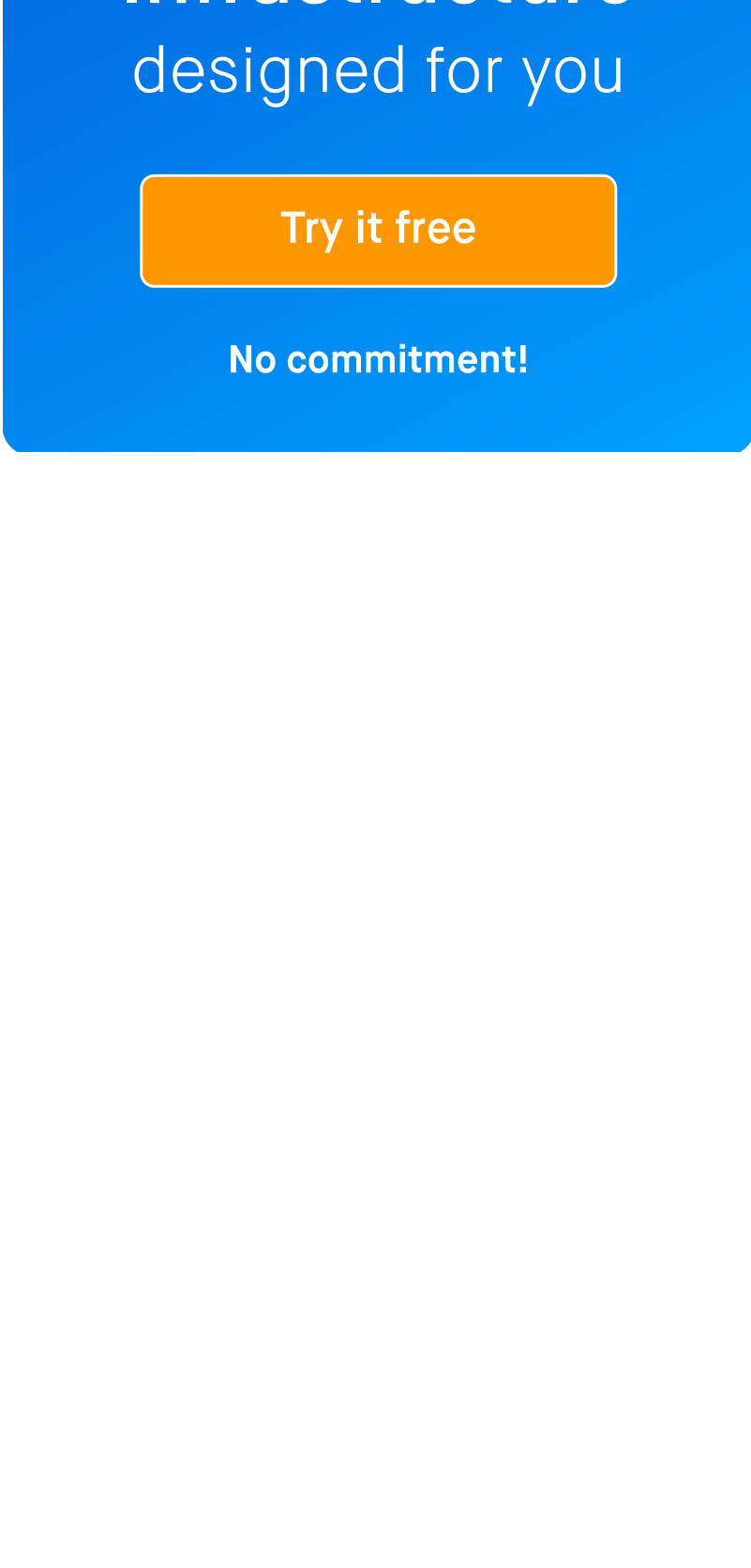
How to Deploy Multi-Node MongoDB Cluster on Ubuntu 20.04

How to install Kubernetes cluster on CentOS 8

How to Install Lighttpd with PHP-FPM and MySQL on Ubuntu 20.04

How To Setup Lumen API on Ubuntu

Set up SSL in the Odoo image



How to Deploy Laravel 8 with Apache and Let's Encrypt SSL on Ubuntu 20.04

Patricia Armesto
1 year ago · Updated

Laravel is a **free, and open-source PHP framework based on the Symfony framework**. It follows the model-view-controller architectural pattern and it's used for the faster development of web applications. It's highly scalable and has built-in support for distributed cache systems. It comes with a command-line interface that helps to perform several operations for your applications.

In this guide, we'll explain **how to install Laravel 8 with Apache and Let's Encrypt SSL on Ubuntu 20.04**.

Requirements

- An Ubuntu 20.04 server installed in your system.
- A valid domain name pointed to your server IP.
- A root password or a user with sudo privileges.

Install Apache Web Server

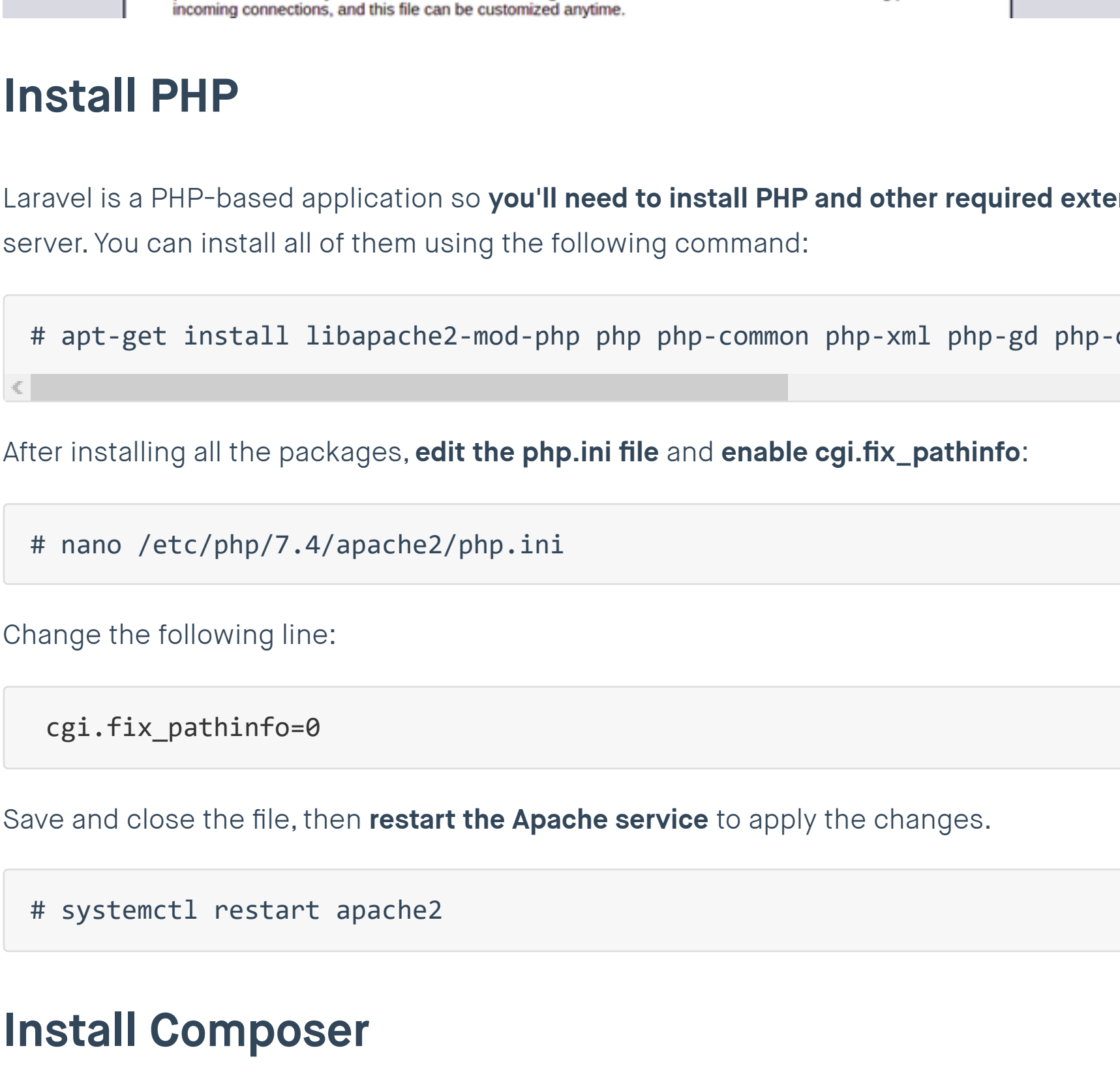
First, you'll need to **install the Web server package** to host a Laravel application. You can use Nginx or Apache web server for Laravel.

To **install the Apache web server**, run the following command:

```
# apt-get install apache2 -y
```

This will install the Apache web server package and start the Apache web service automatically.

To **test the Apache web server**, open your web browser and access the URL **http://your-server-ip**. You should see the Apache test page in the following screen:



Install PHP

Laravel is a PHP-based application so **you'll need to install PHP and other required extensions** to your server. You can install all of them using the following command:

```
# apt-get install libapache2-mod-php php php-common php-xml php-gd php-opcache php-mb
```

After installing all the packages, **edit the php.ini file** and **enable cgi.fix_pathinfo**:

```
# nano /etc/php/7.4/apache2/php.ini
```

Change the following line:

```
cgi.fix_pathinfo=0
```

Save and close the file, then **restart the Apache service** to apply the changes.

```
# systemctl restart apache2
```

Install Composer

Composer is a dependency manager for PHP used for managing dependencies and libraries required for PHP applications.

To **install Composer**, run the following command:

```
# curl -sS https://getcomposer.org/installer | php
```

You should get the following output:

```
All settings correct for using Composer
Downloading...
Composer (version 2.1.6) successfully installed to: /root/composer.phar
Use it: php composer.phar
```

Next, **move the downloaded binary to the system path**:

```
# mv composer.phar /usr/local/bin/composer
```

Next, **verify the Composer version** using the following command:

```
# composer --version
```

You should see the following output:

```
Composer version 2.1.6 2021-08-19 17:11:08
```

Install Laravel 8 Using Composer

In this section, you'll install Laravel using Composer.

First, navigate to the Apache web root directory and **download Laravel** with the following command:

```
# cd /var/www/html
# composer create-project --prefer-dist laravel/laravel laravel
```

You should get the following output:

```
> @php artisan vendor:publish --tag=laravel-assets --ansi
No publishable resources for tag [laravel-assets].
Publishing complete.
@php artisan key:generate --ansi
Application key set successfully.
```

Next, **change the permission and ownership of Laravel directory**:

```
# chown -R www-data:www-data /var/www/html/laravel
# chmod -R 775 /var/www/html/laravel
```

Configure Apache to Host Laravel 8

Next, you'll need to **create an Apache virtual host configuration file to host a Laravel application**.

You can create it using the following command:

```
# nano /etc/apache2/sites-available/laravel1.conf
```

Add the following lines:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName laravel1.example.com
    DocumentRoot /var/www/html/laravel/public

    <Directory /var/www/html/laravel>
        Options Indexes MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Save and close the file and then **enable the Apache rewrite module and activate the Laravel virtual host** with the following command:

```
# a2enmod rewrite
# a2ensite laravel1.conf
```

Finally, **reload the Apache service** to apply the changes:

```
# systemctl restart apache2
```

Install Let's Encrypt SSL on Laravel Site

It's always recommended to **secure your website with Let's Encrypt SSL**. To do so, you'll need to **install the Certbot client to your server**.

Run the following command to install the Certbot client:

```
# apt-get install python3-certbot-apache -y
```

After installing Certbot, run the following command to **download and install the Let's Encrypt SSL on your Laravel website**:

```
# certbot --apache -d laravel.example.com
```

You'll have to **provide a valid e-mail address and accept the term of service** as shown below:

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to cancel): admin@example.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.

-----
(Y)es/(N)o: Y

Obtaining a new certificate
Performing the following challenges:
http-01 challenge for laravel.example.com
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/laravel-le-ssl.conf
Deploying Certificate to VirtualHost /etc/apache2/sites-available/laravel-le-ssl.conf
Enabling available site: /etc/apache2/sites-available/laravel-le-ssl

<----->
```

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.

```
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

Type "2" and hit "Enter" to **download and install a free SSL certificate** for your domain. Once the installation has been completed successfully.

You should get the following output:

```
Enabled Apache rewrite module
Redirecting vhost in /etc/apache2/sites-enabled/laravel.conf to ssl vhost in /etc/a
laravel-le-ssl.conf

-----
Congratulations! You have successfully enabled https://laravel.example.com

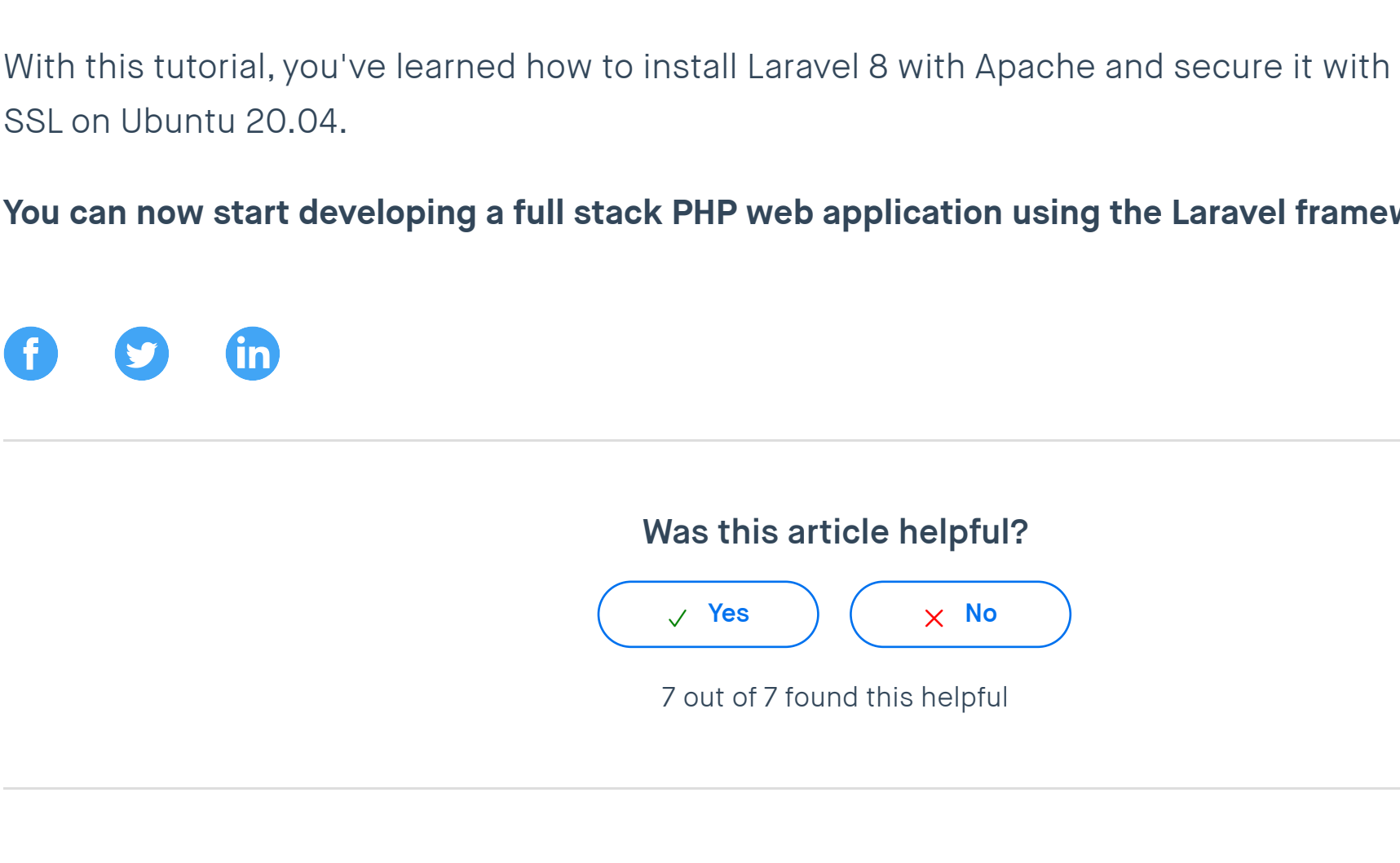
You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=laravel.example.com

<----->
```

At this point, your Laravel website is secured with Let's Encrypt SSL. You can now access it securely using the URL **https://laravel.example.com**.

Access Laravel Website

Now, open your website and type the URL **https://laravel.example.com** in your web browser. You'll be redirected to the Laravel 8 default page:



Conclusion

With this tutorial, you've learned how to install Laravel 8 with Apache and secure it with Let's Encrypt SSL on Ubuntu 20.04.

You can now start developing a full stack PHP web application using the Laravel framework!

[f](#) [t](#) [in](#)

Was this article helpful?

☒ Yes ☐ No

7 out of 7 found this helpful

Related articles

[How to Install Multiple PHP Versions \(7.2, 7.4, 8.0 and 8.1\) on Ubuntu 20.04](#)

[How to Deploy a React.js App with Nginx on Ubuntu 20.04](#)

[How to Install Lighttpd with PHP-FPM and MySQL on Ubuntu 20.04](#)

[How to Deploy a MERN \(MongoDB, Express, React, and Node.js\) Application on Ubuntu 20.04 LTS](#)

[LAMP preinstalled image](#)