

[CHAPTERS](#)[Installing Laravel \[6th Edition\]](#)[Building Our First Website \[6th Edition\]](#)[Building A Support Ticket System \[6th Edition\]](#)[Building A Blog Application \[6th Edition\]](#)[Deploying Our Laravel Applications \[6th Edition\]](#)

Chapter 5: Deploying Our Laravel Applications

Currently, we're just working locally on our personal computers. We will have to deploy our applications to some hosting services or servers so that everyone can access it. There are many ways to make your application visible to the rest of the world!

In this chapter, I will show you how to deploy your Laravel applications using these popular methods:

1. Deploying your apps on shared hosting services
2. Deploying your apps using DigitalOcean

If you find out some better solutions, feel free to contact us. I'll update the book to talk about other approaches.

To deploy a Laravel application, you may have to follow these steps:

- Create a different directory structure. (If you're using shared hosting)
- Setup your web server (If you're using servers or cloud services, such as DigitalOcean, Linode, etc.)
- Upload your applications to your host/server.
- Give proper permissions to your files.
- Create a database for your application.

Deploying your apps on shared hosting services

Basically, your Laravel applications are just PHP applications, which means you can upload them directly to any supported shared hostings, and they may work just fine.

However, Laravel is not designed to work on shared hosting services. Therefore, you will need to do some extra configurations. I don't recommend to use this approach, but the choice is yours.

Here are some popular web hosting services that you can use:

[Host Gator](#)[GoDaddy](#)[Blue Host](#)

Please note that each web hosting service requires a different configuration, so bear in mind that you need to find a way and spend extra time to make your Laravel applications work properly.

Deploying on Godaddy shared hosting

First, let's assume that you have a **new Laravel application** and your **home directory** is:

```
1 | /home/content/learninglaravel/public_html
```

Now, follow these steps:

- You will need to create a directory on the same level as the **public_html** directory. This directory will hold your Laravel application. Because it's on the same level as your **public_html** directory, others cannot access it. This makes the application more secure.
- Upload your **Laravel application** to the new directory using an FTP client (such as Filezilla, Transmit, CuteFTP, etc.), **except the public folder**.
- Upload the **public** folder to the **public_html** directory.

Good job! Now open the **index.php** file and change two paths:

```
1 | require __DIR__.'/../bootstrap/autoload.php';
```

and

```
1 | $app = require_once __DIR__.'/../bootstrap/app.php';
```

Last step, go to the **public_html** directory and modify the **.htaccess** file (If you don't have one, create a new file):

```
1 RewriteEngine On
2 RewriteCond %{REQUEST_URI} !^public
3 RewriteRule ^(.*)$ public/$1 [L]
```

Well done, now it's time to visit your **website URL!** (www.yourdomain.com). You should see the Laravel welcome screen.

Laravel

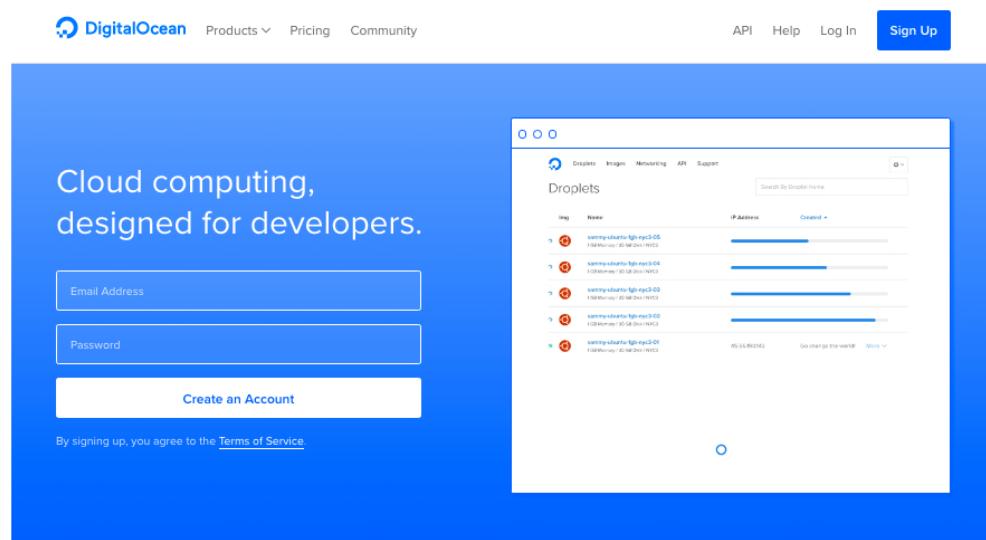
[DOCUMENTATION](#) [LARACASTS](#) [NEWS](#) [FORGE](#) [GITHUB](#)

You've just deployed your Laravel app!

Note: Please note that your application may not work properly on shared hosting and there is a security risk. Laravel is not designed to work on a shared host.

Deploying your apps using DigitalOcean

DigitalOcean is one of the best cloud server providers that you can find around the world. You can get their cheapest SSD Cloud Server for just \$5 a month. More than 450,000 developers have been using DigitalOcean to deploy their applications!



The image shows the DigitalOcean website. On the left, a blue sidebar features the DigitalOcean logo, a 'Sign Up' button, and a 'Log In' button. The main content area has a blue background with the text 'Cloud computing, designed for developers.' and fields for 'Email Address' and 'Password', with a 'Create an Account' button. At the bottom, a small note says 'By signing up, you agree to the [Terms of Service](#)'. On the right, a screenshot of the DigitalOcean dashboard shows a list of 'Droplets' with columns for 'Name', 'IP Address', and 'Created'. The dashboard also includes a chart and a 'Go chart for this world' button.

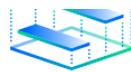
Seamlessly manage your infrastructure





Deploy in seconds

Spin up a Droplet and get root access to a compute instance in only 55 seconds.



SSD performance

The first and only all-SSD cloud. Whether it's our Droplet compute instances or Block Storage, everything runs on SSD.



Simple API

An intuitive API and command line utilities allow you to run large-scale production workloads.



Highly available storage

Never run out of space with the ability to attach multiple highly available volumes up to 16TB to a Droplet.



Lightning fast network

Each hypervisor has a fault tolerant and redundant 40Gbps network to ensure uptime and throughput.



Teams work together

Easily manage your cloud with your team by inviting others and setting access permissions.

DigitalOcean is also my best favorite hosting solution. The performance is really amazing!

In this section, I'll show you how to install Laravel with **Nginx** on a **Ubuntu 16.04.1 LTS VPS**. (Ubuntu 16.04.1 Long Term Support Virtual Private Server)

Why do I choose **Ubuntu 16.04.1**? There are some newer versions of Ubuntu, but the 16.04.1 is an LTS version, which means we will receive updates and support for at least five years.

Alternatively, you can use **Laravel Forge** to install the VPS and configure everything for you. However, you will have to pay \$10/month or more.

Deploy a new Ubuntu server

First, you will need to register a new account at DigitalOcean. You can use this link to get **\$10** for free, that means you can use their **\$5 cloud server for two months**.

<https://www.digitalocean.com/?refcode=5f7e95cb014>

Note: You will need to provide your credit card information or Paypal to activate your account.

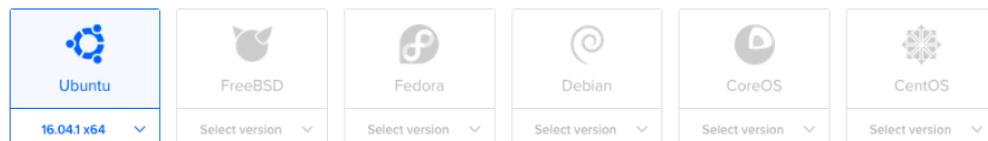
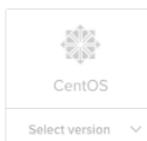
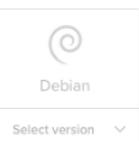
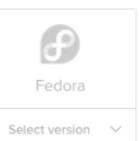
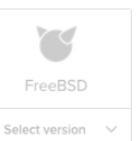
After your account has been activated. You will need to create a "droplet", which is a cloud server. Click on the **Create Droplet** button or go to:

<https://cloud.digitalocean.com/droplets/new>

Create Droplets

Choose an image ?

[Distributions](#) [One-click apps](#) [Snapshots](#)



Choose a size

[Standard](#) [High memory](#)

\$5/mo \$0.007/hour
512 MB / 1 CPU 20 GB SSD disk 1000 GB transfer

\$10/mo \$0.015/hour
1 GB / 1 CPU 30 GB SSD disk 2 TB transfer

\$20/mo \$0.030/hour
2 GB / 2 CPUs 40 GB SSD disk 3 TB transfer

\$40/mo \$0.060/hour
4 GB / 2 CPUs 60 GB SSD disk 4 TB transfer

\$80/mo \$0.119/hour
8 GB / 4 CPUs 80 GB SSD disk 5 TB transfer

\$160/mo \$0.238/hour
16 GB / 8 CPUs 160 GB SSD disk 6 TB transfer

Note: The layout could be different.

Follow these steps:

- At the **Choose an Image** section, be sure to choose **Ubuntu 16.04.1 x64**.
- Select your **droplet size** and region that you like. **\$5/mo** or **\$10/mo** is fine.

- You may skip other settings.
- Give your Droplet a **name** (For example, **learninglaravel**).
- Click "**Create Droplet**" to create your first cloud server!

Wait for a few seconds and...

Congratulations! You just have a new Ubuntu VPS!

Check your email to get the **username** and **password**, you will need to use them to access your server.

```
1 Droplet Name: learninglaravel
2 IP Address: 139.59.245.162
3 Username: root
4 Password: yourPassword
```

Great! Now you can access the new server via **Terminal** or **Git Bash** by using this command:

```
1 ssh root@139.59.245.162
```

Note: Your IP Address should be different

Type **yes** if it asks if you want to continue connecting.

```
~ ssh root@139.59.245.162
The authenticity of host '139.59.245.162 (139.59.245.162)' can't be established.
ECDSA key fingerprint is SHA256:ScoxHEtAHS/81nhm0B6h0w9tMOH9Dj7i/TJ+uWVaF/Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '139.59.245.162' (ECDSA) to the list of known hosts.
root@139.59.245.162's password:
You are required to change your password immediately (root enforced)
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Changing password for root.
(_current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
```

The first time you log in, it will ask you to **change the password**. Enter the **current Unix password** again, and then enter your **new password** to change it.

Finally, run this command to check and update all current packages to the latest version:

```
1 apt-get update && apt-get upgrade
```

We're now ready to install PHP 7, Nginx and other packages!

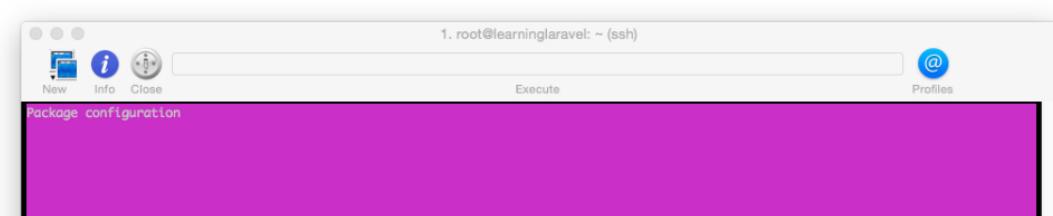
Install MySQL Server

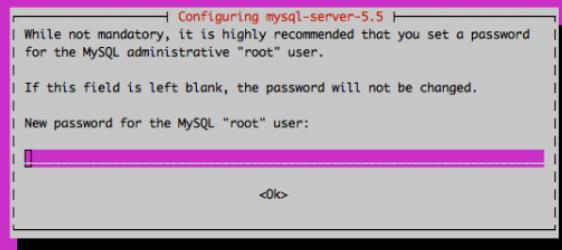
Note: We're using MySQL in this book, so I will install MySQL, you can install other databases if you like.

To get started, we need to install MySQL to store our data. Run this command:

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

Say **Y** to everything. If you're asked to enter a new password for the MySQL root user, give it a password.





Good job! You've installed MySQL. We will install PHP in the next section.

Install Nginx, PHP and other packages

First of all, we need to add **Ondrej's PPA** to the system's **Apt sources** by running this command:

```
1 sudo add-apt-repository ppa:ondrej/php
```

Note: A PPA (Personal Package Archive) is an Apt repository hosted on Launchpad. Third-party developers can distribute their custom PPA packages for Ubuntu outside of the official channels. We have to add the Ondrej's PPA because it **supports PHP 7.0 for Ubuntu**.

Press **Enter (or Return)** to continue if it asks you anything.

Run this command again to update our local packages:

```
1 sudo apt-get update
```

Next, run this command to install **Nginx**, **PHP 7**, **PHP7.0-FPM**, **PHP-MySQL**, **PHP7.0-Zip**, **Curl**, **phpredis**, **xdebug** and other useful packages.

```
1 apt-get -y install nginx php7.0 php7.0-fpm php7.0-mysql php7.0-curl php7.0-xml git php7.0-zip php-redis php-xdebug php7.0-mcrypt
2 php-mbstring php7.0-mbstring php-gettext php7.0-gd
```

Alternatively, you may use this command to install more packages:

```
1 apt-get -y install nginx php7.0-fpm php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-mysql php7.0-phplib
2 php7.0-gd php7.0-imap php7.0-ldap php7.0-pgsql php7.0-pspell php7.0-recode php7.0-tidy php7.0-dev php7.0-intl php7.0-gd
3 php7.0-curl php7.0-zip php7.0-xml git php-redis php-xdebug php7.0-mcrypt php-mbstring php7.0-mbstring php-gettext
```

You may use this command to see all the PHP 7 packages:

```
1 sudo apt-cache search php7-*
```

Available packages:

```
1 php-radius - radius client library for PHP
2 php-http - PECL HTTP module for PHP Extended HTTP Support
3 php-uploadprogress - file upload progress tracking extension for PHP
4 php-mongodb - MongoDB driver for PHP
5 php7.0-common - documentation, examples and common module for PHP
6 libapache2-mod-php7.0 - server-side, HTML-embedded scripting language (Apache 2 module)
7 php7.0-cgi - server-side, HTML-embedded scripting language (CGI binary)
8 php7.0-cli - command-line interpreter for the PHP scripting language
9 php7.0-phplib - server-side, HTML-embedded scripting language (PHPDBG binary)
10 php7.0-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary)
11 libphp7.0-embed - HTML-embedded scripting language (Embedded SAPI library)
12 php7.0-dev - Files for PHP7.0 module development
13 php7.0-curl - CURL module for PHP
14 php7.0-enchant - Enchant module for PHP
15 php7.0-gd - GD module for PHP
16 php7.0-gmp - GMP module for PHP
17 php7.0-imap - IMAP module for PHP
18 php7.0-interbase - Interbase module for PHP
```

```

19  php7.0-intl - Internationalisation module for PHP
20  php7.0-ldap - LDAP module for PHP
21  php7.0-mcrypt - libmcrypt module for PHP
22  php7.0-readline - readline module for PHP
23  php7.0-odbc - ODBC module for PHP
24  php7.0-pgsql - PostgreSQL module for PHP
25  php7.0-pspell - pspell module for PHP
26  php7.0-recode - recode module for PHP
27  php7.0-snmp - SNMP module for PHP
28  php7.0-tidy - tidy module for PHP
29  php7.0-xmlrpc - XMLRPC-EPI module for PHP
30  php7.0-xsl - XSL module for PHP (dummy)
31  php7.0 - server-side, HTML-embedded scripting language (metapackage)
32  php7.0-json - JSON module for PHP
33  php-all-dev - package depending on all supported PHP development packages
34  php7.0-sybase - Sybase module for PHP
35  php7.0-sqlite3 - SQLite3 module for PHP
36  php7.0-mysql - MySQL module for PHP
37  php7.0-opcache - Zend OpCache module for PHP
38  php-apcu - APC User Cache for PHP
39  php-xdebug - Xdebug Module for PHP
40  php-imagick - Provides a wrapper to the ImageMagick library
41  php-ssh2 - Bindings for the libssh2 library
42  php-redis - PHP extension for interfacing with Redis
43  php-memcached - memcached extension module for PHP5, uses libmemcached
44  php-apcu-bc - APCu Backwards Compatibility Module
45  php-amqp - AMQP extension for PHP
46  php7.0-bz2 - bzip2 module for PHP
47  php-rrd - PHP bindings to rrd tool system
48  php-uuid - PHP UUID extension
49  php-memcache - memcache extension module for PHP5

```

* PREMIUM CONTENT: Sorry, this content is available for premium users only.

1 Comment Learning Laravel

 Login

 Recommend  Tweet  Share

Sort by Best



Join the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS 

Name



Realtebo  2 years ago

A question about shared hosting and .htaccess

My .htaccess, in local (development environment) 'public' folder, is

```

<ifmodule mod_rewrite.c="">>
  <ifmodule mod_negotiation.c="">>
    Options -MultiViews
  </ifmodule>
  RewriteEngine On
  # Redirect Trailing Slashes If Not A Folder...
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)/$ /$1 [L,R=301]
  # Handle Front Controller...
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^.* - [E=HTTP_Authorization]
  RewriteRule ^.* - [E=HTTP_Authorization]
</ifmodule>

```

How to modify this ?

   Share >

 Subscribe

 Add Disqus to your site

 Disqus' Privacy Policy

DISQUS



Adobe Creative Cloud for Teams starting at \$29.99 per month. [ads via Carbon](#)