

## Install Laravel on Ubuntu 18.04

First install and setup Apache, PHP ( > or = 7.3 or 8), Mysql/Mariadb. PHP extension 'mbstring' should be enabled on the server.

Then install composer:

```
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer
chmod +x /usr/local/bin/composer
```

Now we can install Laravel:

```
cd /var/www
git clone https://github.com/laravel/laravel.git
cd laravel
```

Now we can install dependency packages of laravel using the command 'composer install'

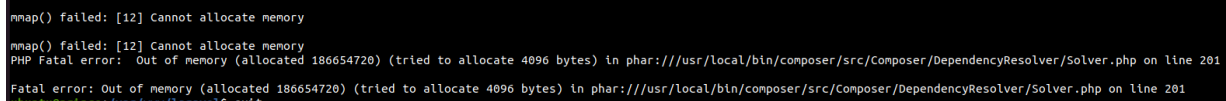
Important Note: Try to avoid running the install command as root user. For more details read:

<https://getcomposer.org/root>

I am running this command as sudo user 'ubuntu'.

```
composer install
```

You may get into some 'memory' issues while running this command.

A terminal window with a black background and white text. It shows several error messages: 'mmap() failed: [12] Cannot allocate memory' appearing twice, followed by 'PHP Fatal error: Out of memory (allocated 186654720) (tried to allocate 4096 bytes) in phar:///usr/local/bin/composer/src/Composer/DependencyResolver/Solver.php on line 201' appearing twice. The terminal prompt is 'ubuntu@ubuntu:~/laravel\$'.

Try running the below command as a workaround:

```
php -d memory_limit=-1 /usr/local/bin/composer install
```

If the issue persists, your RAM is exhausted.

Try creating a swap file:

<https://linuxize.com/post/how-to-add-swap-space-on-ubuntu-18-04>

```

ubuntu@prince:/var/www/laravel$ php -d memory_limit=-1 /usr/local/bin/composer install
Cannot create cache directory /home/ubuntu/.cache/composer/repo/https---repo.packagist.org/, or directory is not writable. Proceeding without cache
Cannot create cache directory /home/ubuntu/.cache/composer/files/, or directory is not writable. Proceeding without cache
Composer is operating significantly slower than normal because you do not have the PHP curl extension enabled.
No composer.lock file present. Updating dependencies to latest instead of installing from lock file. See https://getcomposer.org/install for more information.
Loading composer repositories with package information
Updating dependencies
Lock File operations: 104 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.0.3)
- Locking brick/math (0.9.2)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.8.0)
- Locking facade/ignition (2.9.0)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.14.1)
- Locking fideloper/proxy (4.4.1)
- Locking filp/whoops (2.12.1)
- Locking fruitcake/laravel-cors (v2.0.4)
- Locking graham-campbell/result-type (v1.0.1)
- Locking guzzlehttp/guzzle (7.3.0)
- Locking guzzlehttp/promises (1.4.1)
- Locking guzzlehttp/psr7 (1.8.2)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v8.42.1)
- Locking laravel/sail (v1.6.0)
- Locking laravel/tinker (v2.6.1)
- Locking league/commonmark (1.6.2)
- Locking league/flysystem (1.1.3)
- Locking league/mime-type-detection (1.7.0)
- Locking mockery/mockery (1.4.3)
- Locking monolog/monolog (2.2.0)
- Locking myclabs/deep-copy (1.10.2)
- Locking nesbot/carbon (2.48.0)
- Locking nikic/php-parser (v4.10.5)
- Locking nunomaduro/collision (v5.4.0)
- Locking opis/closure (3.6.2)
- Locking phar-io/manifest (2.0.1)
- Locking phar-io/version (3.1.0)
- Locking phpdocumentor/reflection-common (2.2.0)
- Locking phpdocumentor/reflection-docblock (5.2.2)
- Locking phpdocumentor/type-resolver (1.4.0)
- Locking phpoption/phpooption (1.7.5)
- Locking phpspec/prophecy (1.13.0)
- Locking phpunit/php-code-coverage (9.2.6)
- Locking phpunit/php-file-iterator (3.0.5)
- Locking phpunit/php-invoker (3.1.1)
- Locking phpunit/php-text-template (2.0.4)
- Locking phpunit/php-timer (5.0.3)
- Locking phpunit/phpunit (9.5.4)
- Locking psr/container (1.1.1)

```

After successful installation of dependencies, give the proper permissions to the file:

```
chown -R www-data:www-data /var/www/laravel
```

Set 644 on files:

```
find /var/www/laravel/ -type f -exec chmod 644 {} \;
```

Set 755 on directories:

```
find /var/www/laravel/ -type d -exec chmod 755 {} \;
```

Give full permission on storage directory:

```
chmod -R 777 /var/www/laravel/storage
```

Setup Encryption Key:

```
mv .env.example .env
```

Now generate base64 random number encryption key, using the below command:

php artisan key:generate

```
root@prince:/var/www/laravel# php artisan key:generate
Application key set successfully.
root@prince:/var/www/laravel#
```

Edit the `.env` configuration file and update the APP\_URL:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:ZmPIUjh0VefE7gVD41DjuEJWSv6Q5Wz52tWmqVOCpuY=
APP_DEBUG=true
APP_URL=http://lara.spotfixcrew.com
```

Create Database for Laravel:

```
CREATE DATABASE lara;
GRANT ALL ON lara.* to 'lara'@'localhost' IDENTIFIED BY 'passwd';
FLUSH PRIVILEGES;
```

Again edit the `.env` configuration file and update the db details:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=lara
DB_USERNAME=lara
DB_PASSWORD=passwd
```

Apache Virtualhost:

Disable the default virtualhost:

vim /etc/apache2/sites-enabled/lara.conf

```
<VirtualHost *:80>
```

```
    ServerName lara.spotfixcrew.com
    DocumentRoot /var/www/laravel/public
```

```
    <Directory /var/www/laravel>
        AllowOverride All
    </Directory>
```

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

```
</VirtualHost>
```

Restart Apache:

```
Systemctl restart apache2
```

That's it! Now we can access the Laravel installation from browser:

