

## Deploy de um projeto Laravel 5.7 no Ubuntu 18.04

### 1) Ambiente

Todos os comandos abaixo devem ser executados na máquina virtual.

```
sudo apt update  
sudo apt install apache2
```

Teste a partir de seu navegador, colocando o ip externo de sua VM. Já deve aparecer a página default do apache.

Instalando pacotes do php, dependências do laravel e mysql:

```
sudo apt install php7.2 libapache2-mod-php7.2 php7.2-mbstring php7.2-xmlrpc  
php7.2-soap php7.2-gd php7.2-xml php7.2-cli php7.2-zip  
  
sudo apt install mysql-server-5.7 php-mysql php7.2-mysql mysql-client-core-5.7
```

Configuração do mysql. Pode configurar como desejar os níveis de segurança.

```
sudo mysql_secure_installation
```

Digite:

```
sudo mysql
```

Crie um database e também um usuário para esse database. No caso abaixo, o usuário é "laraveluser" e a senha é "laravel1010@!" :

```
mysql> create database meuprojetodb;  
Query OK, 1 row affected (0.00 sec)  
mysql> GRANT ALL ON meuprojetodb.* TO 'laraveluser'@'localhost' IDENTIFIED BY  
'laravel1010@!';  
Query OK, 0 rows affected, 1 warning (0.00 sec)  
mysql> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
mysql> quit  
Bye
```

Instalando o composer:

```
curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer
```

## 2) Projeto Laravel

Criando um projeto Laravel:

```
cd ~  
composer create-project laravel/laravel meuprojeto --prefer-dist  
cd meuprojeto  
  
# Verifique a versao  
php artisan --version
```

Alterando as configuracoes do .env:

```
nano .env
```

Alterar suas credenciais do mysql =>

```
DB_DATABASE=meuprojetodb  
DB_USERNAME=laraveluser  
DB_PASSWORD=laravel1010@!
```

*Para salvar:*

*ctrl + x*

*Y (ou S se for em português)*

*Enter*

Habilite a autenticao no projeto para testar o mysql:

```
php artisan make:auth  
php artisan migrate
```

## 3) Apache

Agora copie a pasta do projeto para a pasta do apache

```
cd ..  
sudo cp -rp meuprojeto/ /var/www/html
```

Arrume as permissões

```
sudo chown -R www-data:www-data /var/www/html/meuprojeto/  
sudo chmod -R 755 /var/www/html/meuprojeto/
```

Criando uma configuração de entrada no apache para seu novo projeto:

```
sudo nano /etc/apache2/sites-available/meuprojeto.conf
```

Coloque o seguinte conteúdo no arquivo:

```
<VirtualHost *:80>  
    DocumentRoot /var/www/html/meuprojeto/public  
    ServerName seudominio.com  
  
    <Directory /var/www/html/meuprojeto/public>  
        Options +FollowSymlinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

(Caso voce já tenha um dominio que está sendo direcionado para sua VM, coloque-o no arquivo também substituindo o “seudominio.com”)

Configure o serviço do apache para ter certeza q está habilitado

```
sudo systemctl stop apache2.service  
sudo systemctl start apache2.service  
sudo systemctl enable apache2.service
```

Passos finais:

```
# Habilite o modulo de rewrite
sudo a2enmod rewrite

# Habilite a configuracao de seu projeto
sudo a2ensite meuprojeto.conf

# Desabilite a configuracao default (pagina padrao do apache)
sudo a2dissite 000-default.conf

# Restart no servico do Apache
sudo systemctl restart apache2.service
```

Agora coloque novamente o IP externo de sua VM no navegador, ou o seu dominio caso tenha e teste.

---

Caso já tenha um projeto Laravel, será necessário copia-lo para a sua VM Linux ou simplesmente cloná-lo caso esteja em um repositório.

A partir do projeto na VM, instale as dependências:

```
composer install
```

Crie as chaves de segurança:

```
php artisan key:generate
```

Caso ainda não exista nenhum arquivo .env, copie-o do .env.example

```
cp .env.example .env
```

A partir daqui, pode-se seguir os passos a partir da Seção 2 para a configuração do projeto e apache.