
Deploy to Ubuntu VM

The purpose of this guide is to walk you through the setup and deploy of your project on an Ubuntu 22.04 VM.

1. Pre-Tasks

1.1 SSH Connection

You will need to have the SSH key to connect via SSH. If you are using the Amazon EC2, you already had to create a key pair to launch the instance.

Connection example:

```
ssh -i "ssh-key.pem" ubuntu@13.38.15.77
```

All commands shown are to be done on an SSH connect to the VM, unless specifically stated.

Start by updating the local environment:

```
sudo apt-get update
sudo apt-get upgrade
```

The next step is to create the base structure for our project files. This guide assumes the following structure:

```
/
|__var
|__|__www
|__|__|__project
|__|__|__|__laravel
|__|__|__|__vue
```

Run the following command to create that infrastructure

```
sudo mkdir -p /var/www/project/{laravel,vue}
```

2. PHP

First, we need to install PHP and Laravel's required PHP modules.

```
sudo apt-get install -y php7.4 php7.4-
{cli,fpm,common,dev,zip,mbstring,xml,apcu,gd,curl,mysql}
```

Then the composer package manager.

```
sudo php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
sudo php composer-setup.php
sudo php -r "unlink('composer-setup.php');"

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

If the command above gives an error, get the command from its source [here](#).

3. MySQL

Now for the database. Install the MySQL server.

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

Then run the MySQL security setup script, not required but recommended.

```
sudo mysql_secure_installation
```

And provide the following answers:

- Validate Password Component: your choice, yes means passwords need to comply with minimum complexity requirements (and you need to change the root password from the command below);
- Define root password: again, your choice of password;
- Remove anonymous users: Yes;
- Disallow root login remotely: Yes (we'll show you how to connect remotely);
- Remove Test Databases: Yes;
- Reload Privileges: Yes.

The MySQL version on Ubuntu's repositories is 8.0+, and it has a default authentication mechanism that PHP cannot communicate with, so we need to change it. Run the following command.

```
sudo mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'newpassword';"
```

3.1 Check

To check that everything is working run the following commands:

```
mysql -u root -p
```

And inside the MySQL shell.

```
CREATE DATABASE userconsumption;
```

If the database creation command worked everything is ready.

4. Nginx

The last piece of the puzzle is Nginx. This is going to be our Web Server (for the Vue Frontend) and Reverse Proxy (for the Laravel Backend).

Let's install it.

```
sudo apt-get install nginx
```

On the Ubuntu Server, Apache (another web server) is installed by default, so the next step is to disable it and enable Nginx.

```
sudo systemctl stop apache2  
sudo systemctl disable apache2  
sudo systemctl enable nginx  
sudo systemctl start nginx
```

4.1 Check

To check if everything is OK run the following commands and open your VM's IP address on a web browser.

```
sudo systemctl status nginx
```

5. Deploying Code

5.1 Vue Deploy

Before we start copying code, we will configure the permissions on our folders. Run these commands on the server.

```
sudo usermod -a -G www-data ubuntu  
sudo chown -R www-data:www-data /var/www/project  
sudo chmod -R a+rw /var/www/project
```

Compile the build version on your dev machine

```
npm run build
```

Afterwards, use FileZilla to transfer the content of /dist, localized inside vue project, to the VM via SFTP.

5.2 Laravel Deploy

On the server we need to include the dependencies and create the database. Check the values of the local .env (database and root with be dad) and run these commands on the laravel folder in the server.

```
composer install  
php artisan migrate  
php artisan passport:install
```

Copy the client secret to your .env file.

5.3 Nginx Configuration

By default, PHP FPM (which is what we are using as a PHP "server") runs as an UNIX socket, so our config file for Nginx is going to be the following.

```
server {  
    listen 80;  
  
    server_name 15.188.51.61 default;  
  
    root /var/www/project/laravel/public;  
  
    charset utf-8;  
  
    location ~ ^/(api|storage|oauth)/.* {  
        root /var/www/project/laravel/public;  
        try_files $uri /index.php?$is_args$args;  
    }  
  
    location ~ \.php$ {  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
        fastcgi_index index.php;  
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;  
        include fastcgi_params;  
    }  
  
    location / {  
        root /var/www/project/vue;  
        index index.html;  
        try_files $uri $uri/ /index.html;  
    }  
  
    location ~ /\.(!well-known).* {  
        deny all;  
    }  
}
```

Create a file on /etc/nginx/sites-available named sem with the contents above.

```
sudo nano /etc/nginx/sites-available/sem
```

Setup, check and restart Nginx

```
sudo rm /etc/nginx/sites-enabled/default  
sudo ln -s /etc/nginx/sites-available/sem /etc/nginx/sites-enabled/sem  
sudo nginx -t  
sudo systemctl restart nginx
```