Deploying Laravel-Vue.js app on AWS EC2 Instance (Ubuntu)

Steps:

1. Create EC2 Instance
2. Install Nginx web server
3. Install PHP
4. Install Composer
5. Install Node
6. Install MySQL
7. Install PHPMyAdmin
8. Clone GitHub
9. Setup Domain and Subdomains
10. Prepare Laravel Project
11. Prepare Vuejs Project
12. Setup SSL

1. Create EC2 Instance

Create EC2 instance for Ubuntu. Also create .pem key pair file while creating instance. Allow ssl, http, https traffic in network settings.

To connect instance from any ssh client, use following command:

```
ssh -i "<filename>.pem@<public ip or public dns>
```

If public IP doesn't load in browser, it is better to use elastic ip.

2. Install Nginx web server

```
sudo apt update
sudo apt upgrade
sudo apt install nginx
```

3. Install PHP

```
sudo apt install ca-certificates apt-transport-https software-properties-common
sudo add-apt-repository ppa:ondrej/php

sudo apt update
sudo apt upgrade
sudo apt install php8.1 -y

sudo apt install php8.1-fpm php8.1-mysql php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-zip php8.1-curl unzip   php8.1-gd

sudo systemctl restart nginx.service
```

4. Install Composer

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

5. Install Node

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash

. ~/.nvm/nvm.sh

nvm install --lts
```

6. Install MySQL

```
sudo apt update
sudo apt install mysql-server
sudo mysql

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your password';
```

7. Install PHPMyAdmin

```
sudo apt update
sudo apt install phpmyadmin
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

  - Check Permissions:
  sudo chown -R www-data:www-data /var/www/html/phpmyadmin
  sudo chmod -R 755 /var/www/html/phpmyadmin
  sudo chown -R www-data:www-data /var/www/html


cd /etc/nginx/sites-available/
sudo nano default

// Add these lines

  location /phpmyadmin {
      root /var/www/html;
      index index.php index.html index.htm;

      location ~ ^/phpmyadmin/(.+\.php)$ {
          try_files $uri =404;
          root /var/www/html;
          fastcgi_pass unix:/run/php/php8.1-fpm.sock;
          fastcgi_index index.php;
          fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
          include /etc/nginx/fastcgi_params;
      }

      location ~* ^/phpmyadmin/(.+\.(jpg|jpeg|gif|css|png|js|ico|html|xml|txt))$ {
          root /var/www/html;
      }
  }
```

8. Clone GitHub

```
sudo apt update
sudo apt install openssh-client
```

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id_rsa

cat ~/.ssh/id_rsa.pub
```

Use the SSH key in github account. Alternatively you can create a personal access token and use it instead of password while cloning the app through https.

```
cd /var/www/vhosts
sudo git clone <git url>

sudo mv <git folder name> mydomain.com
sudo chown ubuntu:ubuntu -R mydomain.com
```

9. Setup Domain and Subdomains and change nginx config

Setup cname, A record for domain name and also create a subdomain for backend using another A record. Eg. backend.mydomain.com

Then in ssh terminal, run these commands

```
cd /etc/nginx/sites-available/
sudo nano backend
```

On backend file, paste these lines:

```
server {
    listen 80;
    listen [::]:80;
    server_name backend.mydomain.com;
    root /var/www/vhosts/mydomain.com/backend/public;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";

    index index.php;
    charset utf-8;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt  { access_log off; log_not_found off; }

    error_page 404 /index.php;

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.(?!well-known).* {
        deny all;
    }
}
```

Then run following commands to create symbolic link

```
cd ../sites-enabled/
sudo ln -s ../sites-available/backend .
ls -l
```

Also change default so that it loads vuejs frontend

```
sudo nano default
```

Change rooth path like this:

```
root /var/www/vhosts/mydomain.com/frontend/dist;
```

```
sudo service nginx restart
```

10. Prepare Laravel Project

```
cd /var/www/vhosts/mydomain.com/backend
sudo chown www-data:www-data -R storage
sudo chown www-data:www-data -R bootstrap/cache

composer update
sudo cp .env.example .env
sudo nano .env

sudo php artisan key:generate
php artisan migrate
```

11. Prepare Vue.js Project

It is better to build project locally using npm run build or vite build command (if using vite). We have set nginx root folder path to ‘dist’ folder in step 9 above. It should be enough to load the site

12. Setup SSL

```
sudo snap install core;
sudo snap refresh core

sudo snap install --classic certbot

sudo ln -s /snap/bin/certbot /usr/bin/certbot

sudo certbot --nginx -d mydomain.com -d www.mydomain.com -d backend.mydomain.com


sudo systemctl status certbot.timer

sudo certbot renew --dry-run
```