Laravel DEPLOY ON AWS

STEP 1: CLONE PROJECT

we need to go `/var/www/html` and then use `git clone your_reposity` to clone the project.

```
cd /var/www/html
git clone your_reposity
```

STEP 2: `ENVIROMENT VARIABLE` FILE

In Laravel when we push the project to GitHub we ignore the `.env` file but we still have `.env.example` file.

You just copy `.env.example` and rename to `.env` by following command in linux:

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

STEP 3: INSTALL THE PHP LIBRARY

```
composer install
```

To install the PHP Library in `vendor` folder

STEP 4: INSTALL THE JS LIBRARY

```
npm install
```

OR

```
npm i
```

To install JavaScript Library in `node_modules` folder

STEP 5: `CREATE DATABASE`

You need to create the database and config in `.env` file on `DB_DATABASE= your_db_name` example:

```
DB_DATABASE=example_db
DB_USERNAME=root
DB_PASSWORD=123
```

STEP 6: `LARAVEL KEY GENERATOR`

When we copy `.env.example` file and rename to `.env` file in `APP_KEY=` is empty so you cannot run your Laravel project because in security purpose.

So to fix this issue we need to run the following command:

```
php artisan key:generate
```

STEP 7: `MIGRATE TABLE TO DATABASE`

Following the step above now we can run migrations command to migrate our table to database.

> php artisan migrate

For some reason with your Database instance Maria-DB version you might be meet the error like *Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes* . To fix this issue you need to add some code on `AppServiceProvider.php`.

> app/Providers/AppServiceProvider.php

```
use Illuminate\Support\Facades\Schema; // add this line
public function boot()
{
    Schema::defaultStringLength(191); // add this line
}
```

After that you need to delete all your table in your database and then run your migration again: `php artisan migrate`

STEP 8: GIVE PERMISSIONS TO `storage` folder.

When we are on Server we don't have permission to access the storage folder so we need to give the permission to user by following the command:

> sudo chmod 777 -R storage

STEP 10: `ALLOW OVERRRIDE`

After we finish step 6 we still cannot access the path on URL or link to other pages we need to to allow override in `httpd.conf` file.

> sudo vi `/etc/httpd/conf/httpd.conf`

Press `i` on keyboard to insert or edit file and then scroll down to find:

```
# Purther relax access to the default document root:
<Directory "/var/www/html">
.....................................................................
.....................................................................
AllowOverride None
```

> AllowOverride `None`

Update `None` to `All`

> AllowOverride `All`

After that we need to restart the server by running the following command:

> sudo `systemctl restart httpd.service`

STEP 11:

You might be have a problem with images when upload and cannot display this problem because of `systemlink` so you need to put your image in `public` folder and also you need to run the following command:

> php artisan storage:link