

Steps-

1. Create a new EC2 instance with **Ubuntu OS** and configure the inbound rules in the security group of the instance to allow all necessary access.

Security group rule...	IP version	Type	Protocol	Port range
sgr-01d69c0575125d...	IPv4	SSH	TCP	22
sgr-0df5e1c28a4753650	IPv4	HTTPS	TCP	443
sgr-0ff8e91eaeacca604	IPv4	Custom TCP	TCP	8000
sgr-001307bff7a5a4020	IPv4	HTTP	TCP	80

2. Now setup the server with Nginx, MySql , and the required version of PHP.
3. Login to the server ssh terminal using the pem file.
4. Run the following commands to install nginx-
 - a. `sudo apt update`
 - b. `sudo apt install nginx`
 - c. `sudo systemctl restart nginx`
5. After completing the steps, you should be able to access the Nginx page by entering your IP address in a web browser, which will display the "Welcome to Nginx" message.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

6. We are using PHP 8.1 in this project.
7. Run the following commands to install PHP 8.1 and all required php components-
 - a. `sudo apt update`
 - b. `sudo apt install --no-install-recommends php8.1`
 - c. `sudo apt-get install -y php8.1-cli php8.1-common php8.1-mysql php8.1-zip php8.1-gd php8.1-mbstring php8.1-curl php8.1-xml php8.1-bcmath php8.1-fpm`
 - d. `curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php`
 - e. `HASH=`curl -sS https://composer.github.io/installer.sig``
 - f. `echo $HASH`
 - g. `php -r "if (hash_file('SHA384', '/tmp/composer-setup.php') === '$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"`
 - h. `sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer`

8. Now all the required components on PHP are installed and you can check the php version using command - "php -v". You will get output something like

```
ubuntu@ip-172-31-2-100:~$ php -v
PHP 8.1.2-1ubuntu2.13 (cli) (built: Jun 28 2023 14:01:49) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.13, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-2-100:~$
```

9. Now we need to install MySQL on the server for database.
10. Steps to install MySQL-
- sudo apt update
 - sudo apt install mysql-server
11. Now to set up a new user we need to run some queries to create our own db and give permissions to the user.
12. To do this we need run few commands-
- sudo mysql
 - CREATE DATABASE emsdb;
 - CREATE USER 'hauper_ems'@'localhost' IDENTIFIED BY 'hauperadminems';
 - GRANT ALL PRIVILEGES ON emsdb.* TO 'hauper_ems'@'localhost';
 - FLUSH PRIVILEGES;
13. The above commands will create a DB named 'emsdb' , Username "hauper_ems" and Password for the user will be "hauperadminems"
14. This will be the db we will use in the .env file of the laravel application.
15. Now we go the directory ' /var/www/html/ ' using cd /var/www/html/

```
ubuntu@ip-172-31-2-100:~$ cd /var/www/html/
ubuntu@ip-172-31-2-100:/var/www/html$ ls
index.nginx-debian.html
ubuntu@ip-172-31-2-100:/var/www/html$
```

16. Now we clone our git repository which has all the code here inside this folder.
17. Now cd into the folder and create a new file named .env using the command ' sudo touch .env '
18. Open this file in a file editor nano or vim and paste all required environment variables in the file.

19. Change the db credentials , App url (DNS name) and other required components.

```
APP_NAME="Hauper EMS"
APP_ENV=local
APP_KEY=base64:l1AbnVrUc2vXfMDALkAWqBe/VuS+9FRIr8n55IMA0es=
APP_DEBUG=true
APP_URL=ec2-3-110-154-28.ap-south-1.compute.amazonaws.com

LOG_CHANNEL=daily
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=emsdb
DB_USERNAME=hauper_ems
DB_PASSWORD=hauperadminems
```

20. Now run few PHP commands on the server -

- sudo composer install
- sudo php artisan key:generate
- sudo php artisan migrate --seed

21. Now change owner of the folder using the command -

- sudo chown -R www-data:www-data /var/www/html.

22. This is the user nginx uses to run the application so this user needs all permissions.

23. Now cd into the directory using 'cd /etc/nginx/sites-available/' and create a file named `ems.conf`

```
ubuntu@ip-172-31-2-100:/var/www/html/hauper-ems$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-2-100:/etc/nginx/sites-available$ ls
default
ubuntu@ip-172-31-2-100:/etc/nginx/sites-available$ sudo touch ems.conf
ubuntu@ip-172-31-2-100:/etc/nginx/sites-available$
```

24. Now using your editor nano or vim open the file and paste this script in the conf file.

EMS Nginx script

25. Now enable this script using the command

- sudo ln -s /etc/nginx/sites-available/ems.conf
/etc/nginx/sites-enabled/ems.conf

26. Now check the configuration file using command

- sudo nginx -t
- Now the output should be something like

```
ubuntu@ip-172-31-2-100:/etc/nginx$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```


27. Now restart the nginx service using

- sudo systemctl restart nginx

28. Now to setup ssl certificate use the following link which is official certbot documentation.

<https://certbot.eff.org/instructions?ws=nginx&os=ubuntu>

29. Select your web server and follow the instructions till number 6



certbot instructions about certbot contribute to certbot hosting providers with https get help donate

certbot instructions

My HTTP website is running Nginx on Ubuntu 20

[Help, I'm not sure!](#)

Nginx on Ubuntu 20

6. Choose how you'd like to run Certbot

Either get and install your certificates...

Run this command to get a certificate and have Certbot edit your nginx configuration automatically to serve it, turning on HTTPS access in a single step.

```
$ sudo certbot --nginx
```

30. Do the following steps till number 6 and it will install the ssl certificate.

31. Now you will be able to access the website using the https link.