                               LMS Project on VIRTUAL MACHINE

**Instance type-t2 medium**

**Os-**ubuntu-20.04

**Security group**  -443,5432,80,8080

* Clone the code into the server

sudo git clone<https://github.com/KonaMarsTech/lms-public.git>

* Change the directory into lms-public/ i.e  cd lms-public/

git checkout vm-docker-cicd

Install [NodeJS](https://nodejs.org/en/) on the server

*Node.js is an open-source, cross-platform JavaScript runtime environment.*

**Command**:

curl -fsSL https://deb.nodesource.com/setup\_16.x | sudo -E bash - &&sudo apt-get install -y nodejs

       -   Got to Web-App

        cd webapp/

       -  Run  the following commands

**Open the Env File**

sudo vi .env

* And edit  your < domainr>

VITE\_API\_URL=https://your\_domain-name/api

       sudo npm install

      sudo npm run build

**Install Nginx :**

sudo apt install nginx

sudo rm /etc/nginx/sites-enabled/default

sudo vi /etc/nginx/sites-available/lms-app

server {

    server\_name your\_domain.com;

    location / {

        root /home/ubuntu/lms-public/webapp/dist;

    }

    location /api {

        proxy\_pass http://localhost:8080;

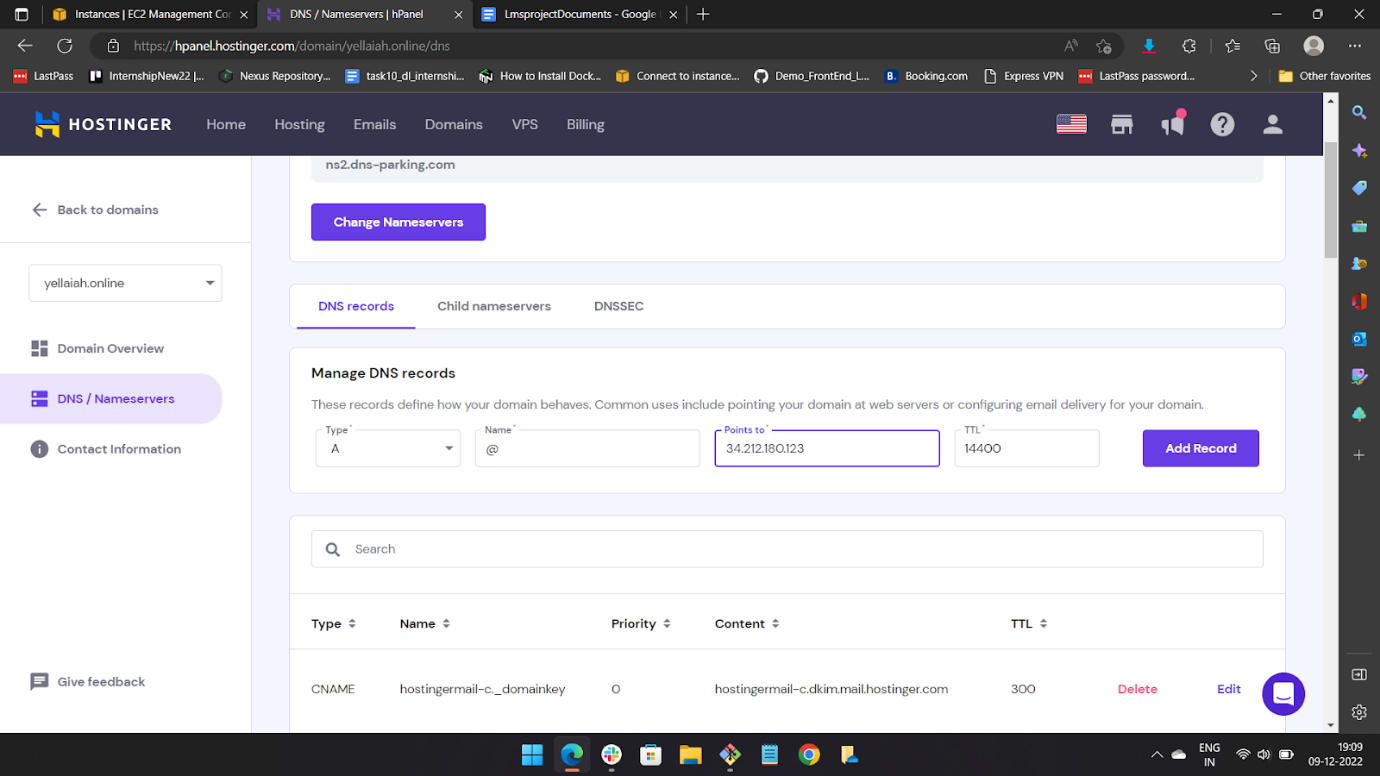
    }

}

* Enable site by the following command

sudo ln -s /etc/nginx/sites-available/lms-app /etc/nginx/sites-enabled/lms-app

Setup A Record in GoDaddy :



Paste your ip address:

Install CertBot : for https connection

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

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

mail id --- any mail id

Yes

Yes

Select the number of your domain

 sudo nginx -t

sudo nginx -s reload

-----------------------------------------------------------------------------------------------

Install Postgres DB

<https://www.postgresql.org/download/>

<https://www.postgresql.org/download/linux/ubuntu/>

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

 wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

sudo apt-get update

sudo apt-get -y install postgresql

sudo su postgres

Type the command – psql

* Set the password for the database by the following command
* **\password**
* And enter your password------- password

Setup Backend Setup

* Change the directory to /api

cd ../api

* And run the below commands to build the code

 sudo npm install

 sudo npm install -g pm2

* And edit the .env file

sudo   vi .env

* Copy the below Content in .env file and give your database <password> in password area

MODE=production

PORT=8080

DATABASE\_URL=postgresql://postgres:**password**@localhost:5432/postgres

 sudo  npx prisma db push

sudo npm run build

* And run the below command

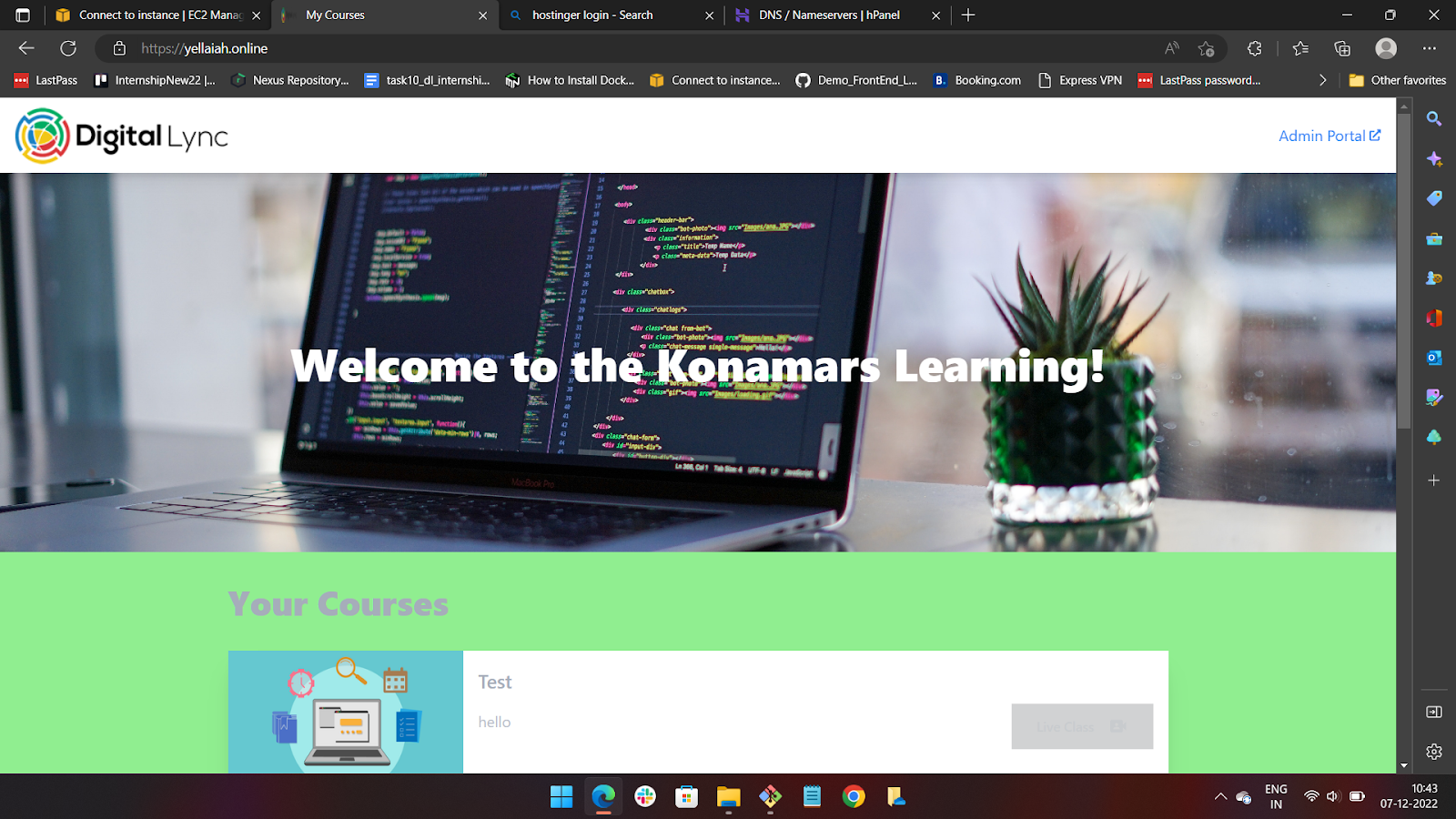
NODE\_PORT=8080 pm2 start -i 0 build/index.js

sudo systemctl restart nginx

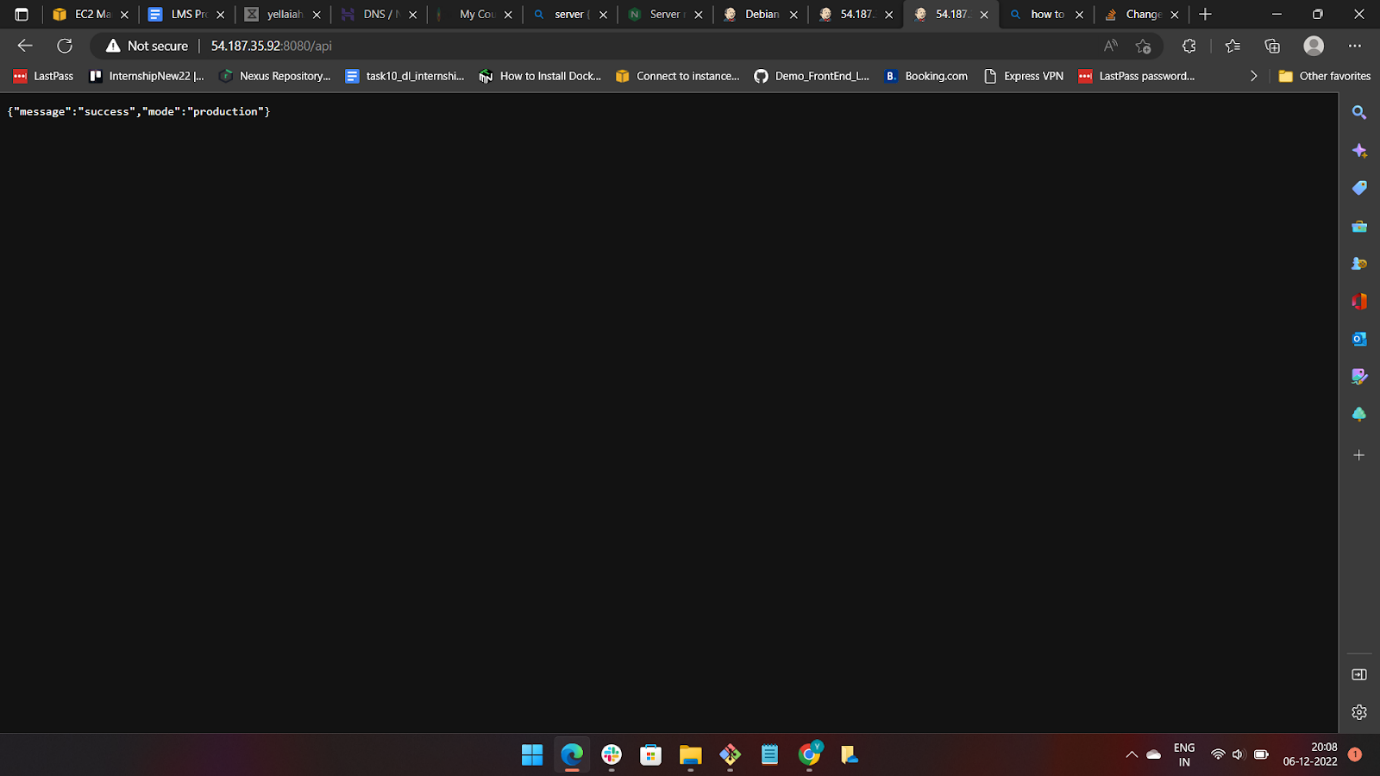
* pm2 list
* pm2 monit
* Check the frontend and Backend connections in CLIby running the below commands

curl http://localhost

 curl<http://localhost:8080/api>

* And check with your domain name in the url
* Frontend page with https connection
* 

Backend successfully configured



***Step-by-Step Process:***

Step1: Create instance as per the required specifications

Step2: Clone the code from github

Step3: Change the Directory to LMS and the git checkout to vm-dockert-cicd

Step4: Install Node.js in the server

Step5: Go to webapp and change the domain name in .env file

Step6: Install Nginx

Step7: Replace the sudo rm /etc/nginx/sites-enabled/default with sudo vi /etc/nginx/sites-available/lms-app and write a code in it, with the domain name in it.

Step8: Enable site by the following command sudo ln -s /etc/nginx/sites-available/lms-app /etc/nginx/sites-enabled/lms-app

Step9: Setup “A” Record in GoDaddy an Paste your ip address in it.

Step10: Install CertBot for https connection

Step11: Install Postgres DB

Step12: Install Backend Setup by Changing the Directory to /api

Step13: Edit the .env file with the following code given above.

Step14: Restart Nginx.