JENKINS DOCKER PIPELINE BACKEND

* **Instance type-t2 medium**
* **Os-**ubuntu-20.04
* **Security group**  -443,5432,80,8080
* Install docker in Jenkins server .Visit the site [**www.get.docker.com**](http://www.get.docker.com)**.**
* curl -f

curl -fsSL https://get.docker.com -o get-docker.sh

sh get-docker.sh

-Run the following commands

sudo systemctl start docker

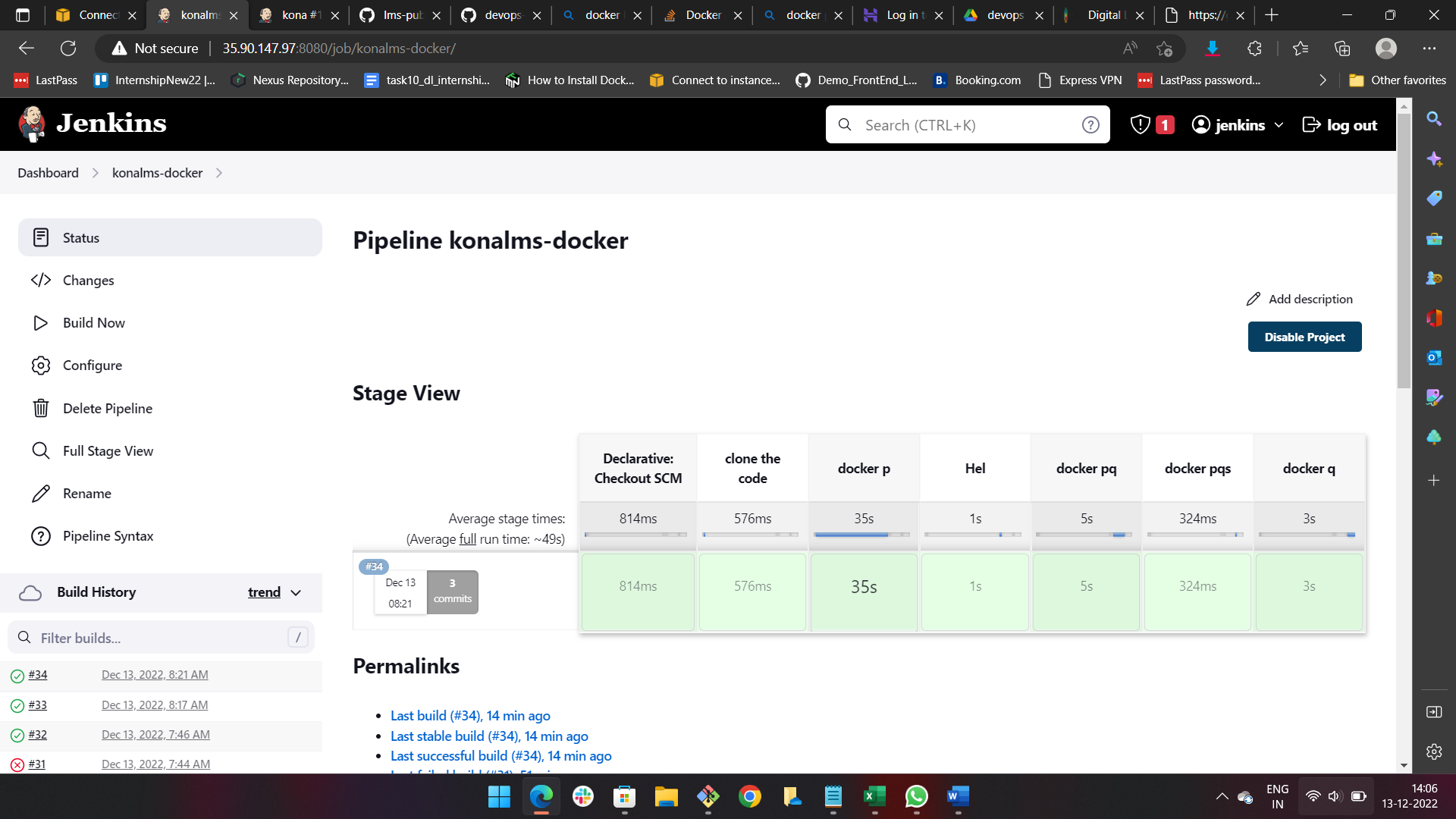
sudo systemctl enable docker

sudo usermod -aG docker centos(i.e,, adding centos user to docker group)

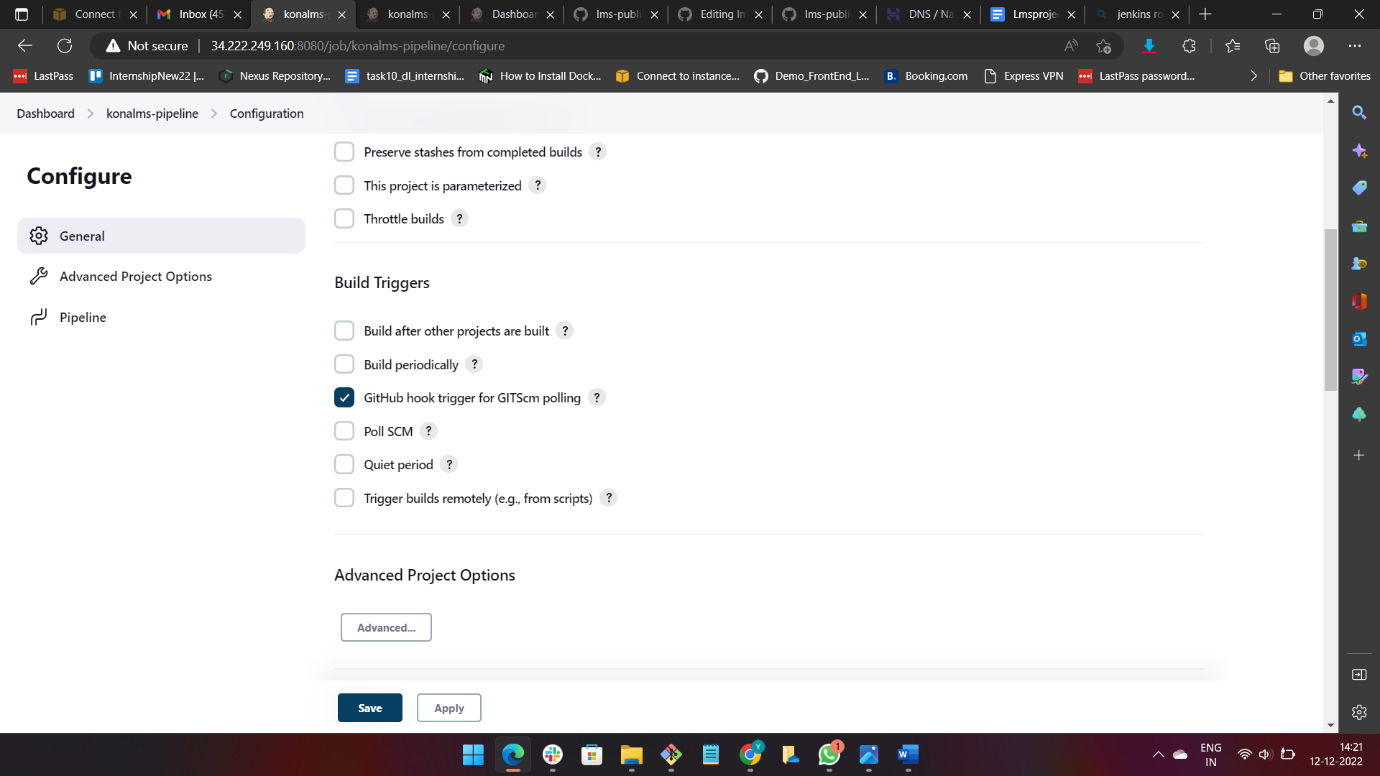
sudo usermod -aG docker jenkins(i.e,,adding jenkins user to docker group)

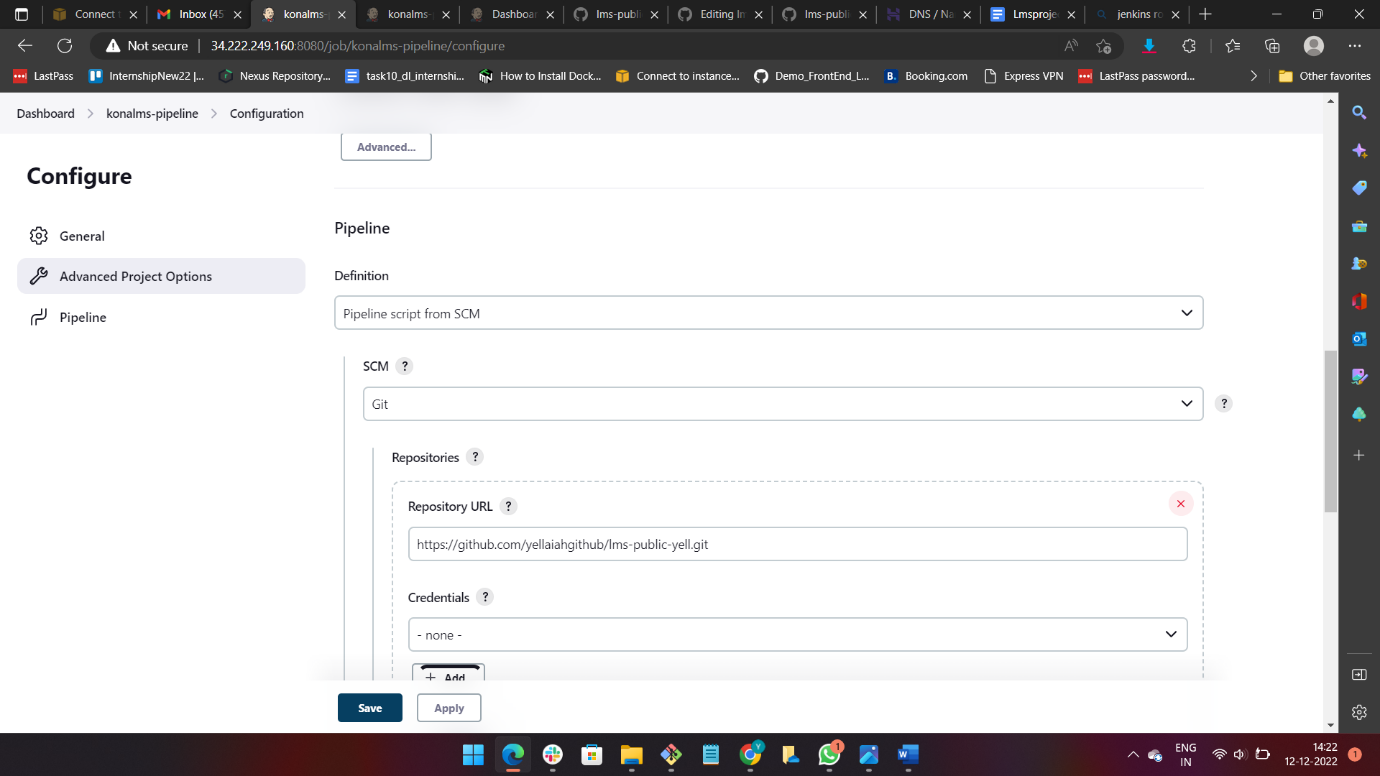
newgrp docker

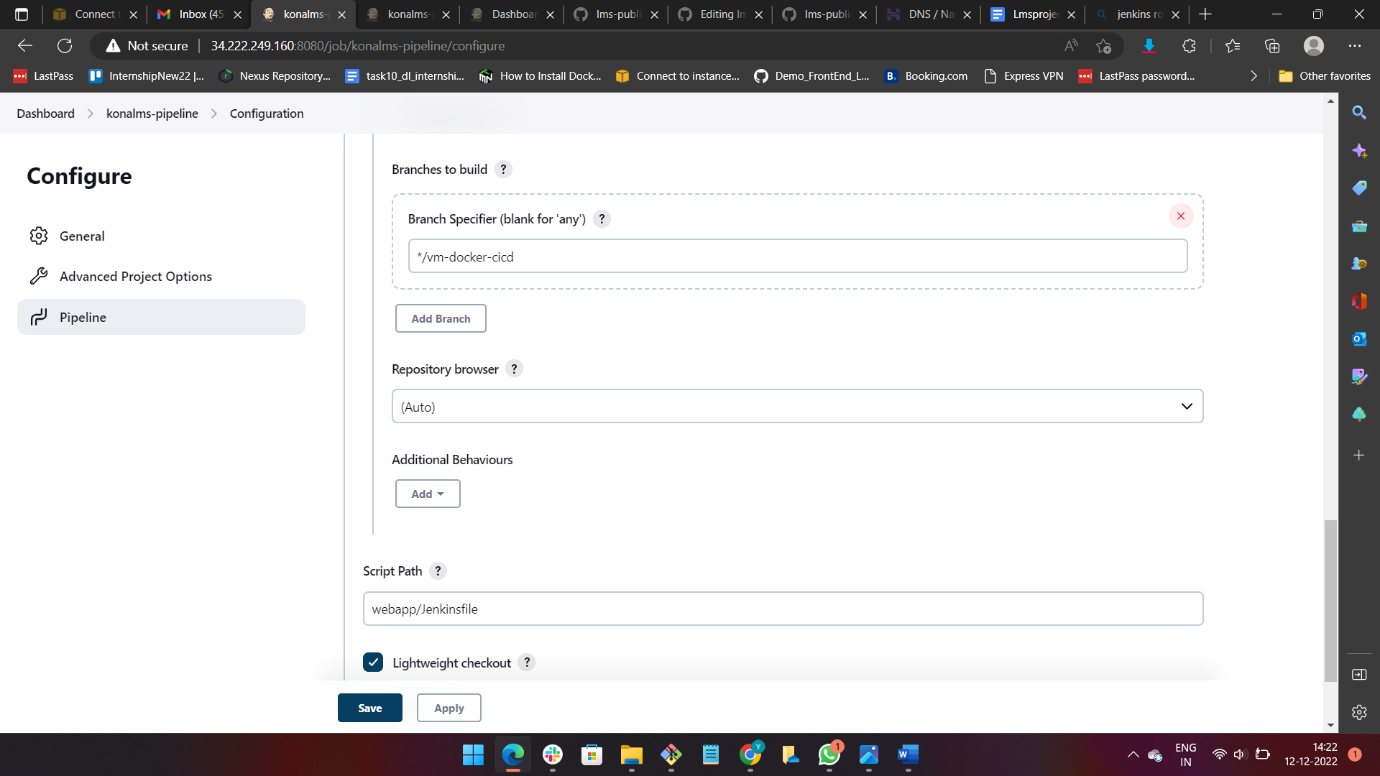
* Now create a konalms-docker job in jenkins

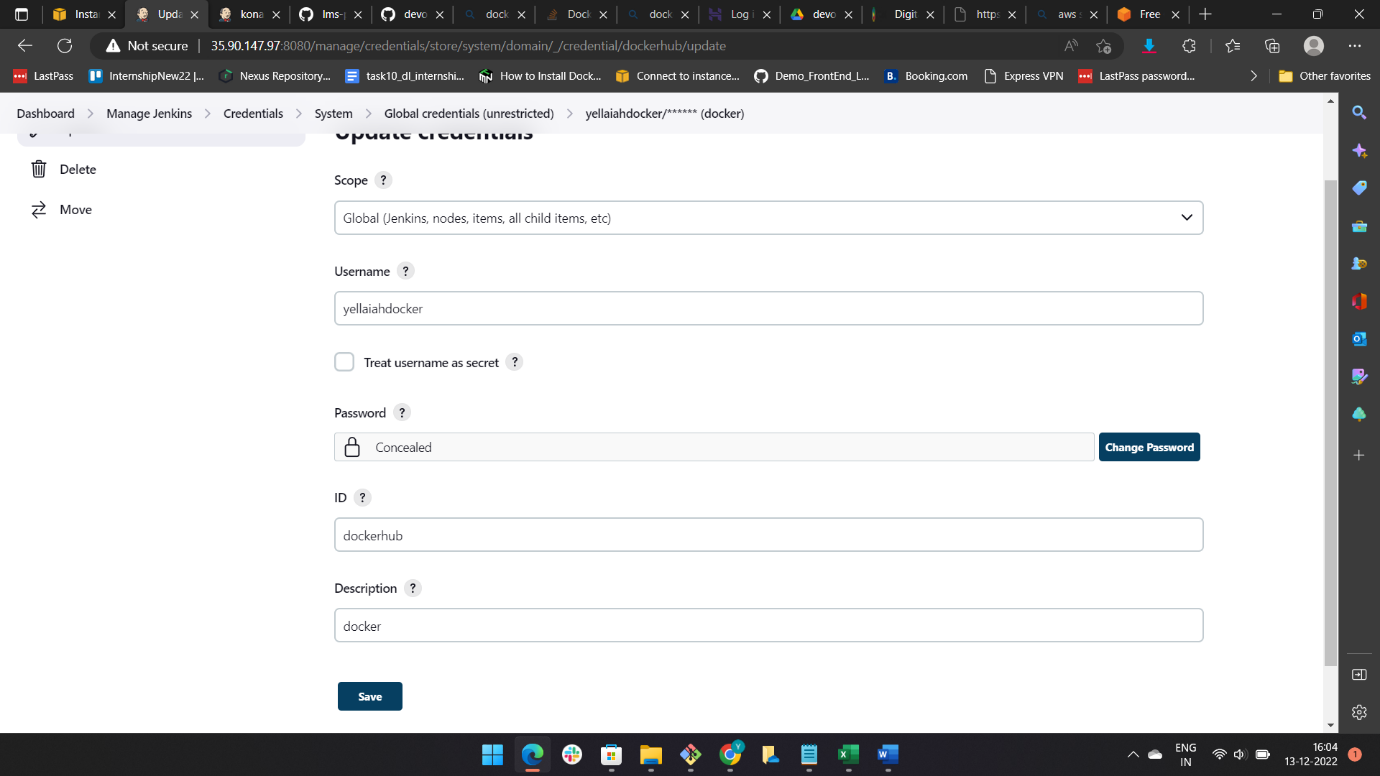


* Now configure the pipeline as shown like below pictures







* Install docker plugins
* **Note**: Give your jenkins file name in the script path i.e
* api/jenkinsfile
* Now add the dockerhub credentials in jenkins
* Go to manage jenkins🡪manage credentials
* Update your credentials with your username and password of dockerhub account as shown below
* 

Note: Give your jenkins file name in the script path i.e api/Jenkinsfile-docker-backend

* Now create a Jenkinsfile-docker-backend file in lms-public/api

pipeline {

agent{

label 'docker'

}

environment {

DOCKERHUB\_CREDENTIALS = credentials('dockerhub')

registry = "yellaiahdocker/backend-lms"

registryCredential = 'dockerhub'

dockerImage = ''

}

stages {

stage('creating new docker network') {

steps {

sh 'docker network create lmsnetwork'

}

}

stage('Building the docker image') {

steps {

sh 'cd api && docker build -t yellaiahdocker/backend-lms .'

}

}

stage('Logging into dockerhub account') {

steps {

sh 'echo $DOCKERHUB\_CREDENTIALS\_PSW | docker login -u $DOCKERHUB\_CREDENTIALS\_USR --password-stdin'

}

}

stage('pushing the docker image into dockerhub') {

steps {

sh 'docker push yellaiahdocker/backend-lms'

}

}

stage('Remove old docker images') {

steps {

sh 'docker rmi -f yellaiahdocker/backend-lms'

}

}

stage('creating database container') {

steps {

sh 'docker container rm --force lmsdb'

sh 'docker run -d -p 5432:5432 --network lmsnetwork -e POSTGRES\_PASSWORD=password --name lmsdb postgres'

}

}

stage('Running the docker container') {

steps {

sh 'docker container rm --force backend'

sh 'docker run -d -p 8080:8080 --network lmsnetwork -e DATABASE\_URL=postgresql://postgres:password@lmsdb:5432/postgres --name backend -e PORT=8080 -e MODE=local yellaiahdocker/backend-lms'

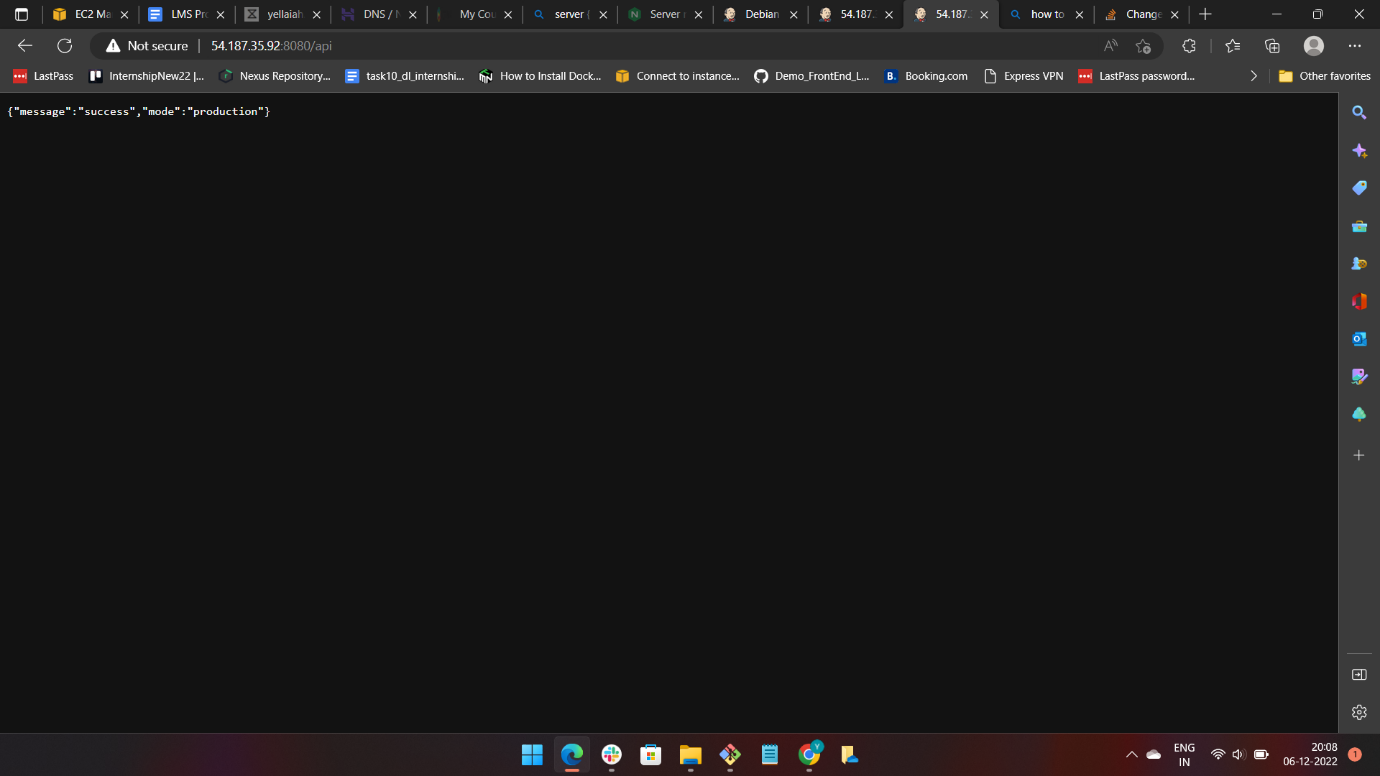
}

}

}

}

* Now build the job in jenkins
* Check the ip address



* Whenever the developer makes the changes ,the jenkins job starts automatically and deployes in the backend docker container.