Project Number – 1
Deploying Dynamic Web Application

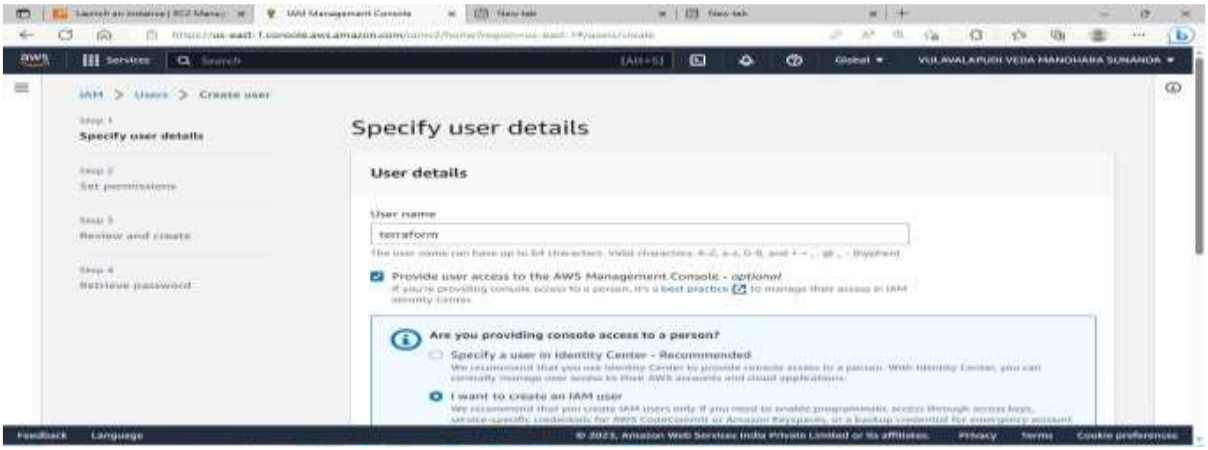
2000031095
Section13
Cloud and Devops

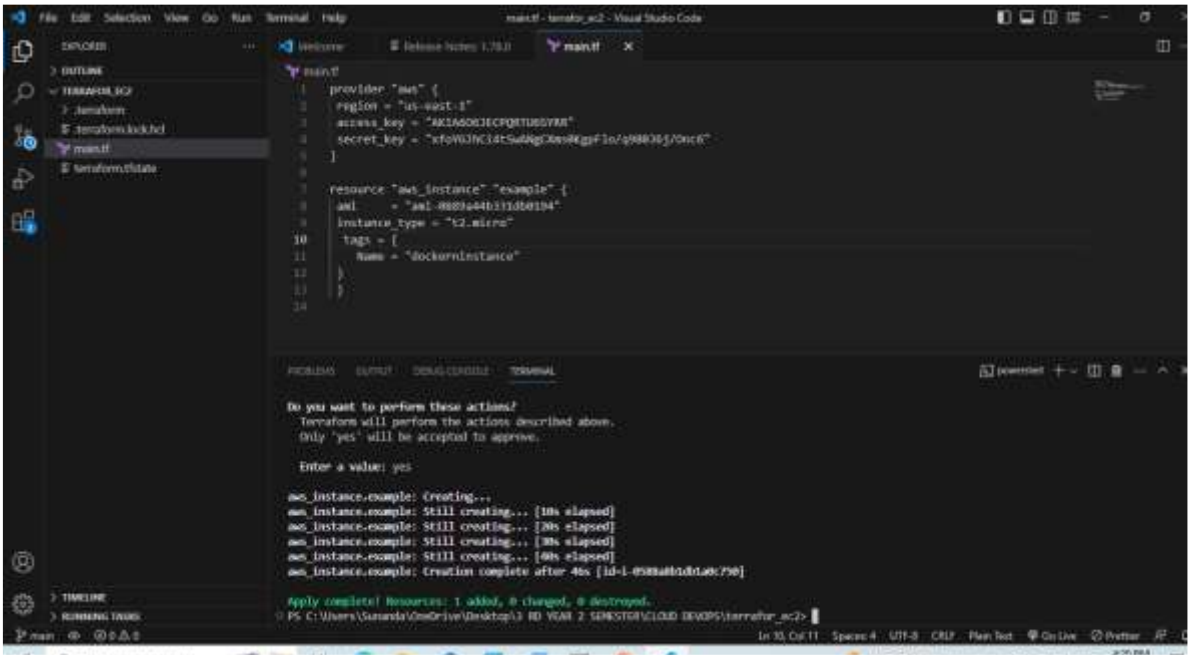
Services I have used are

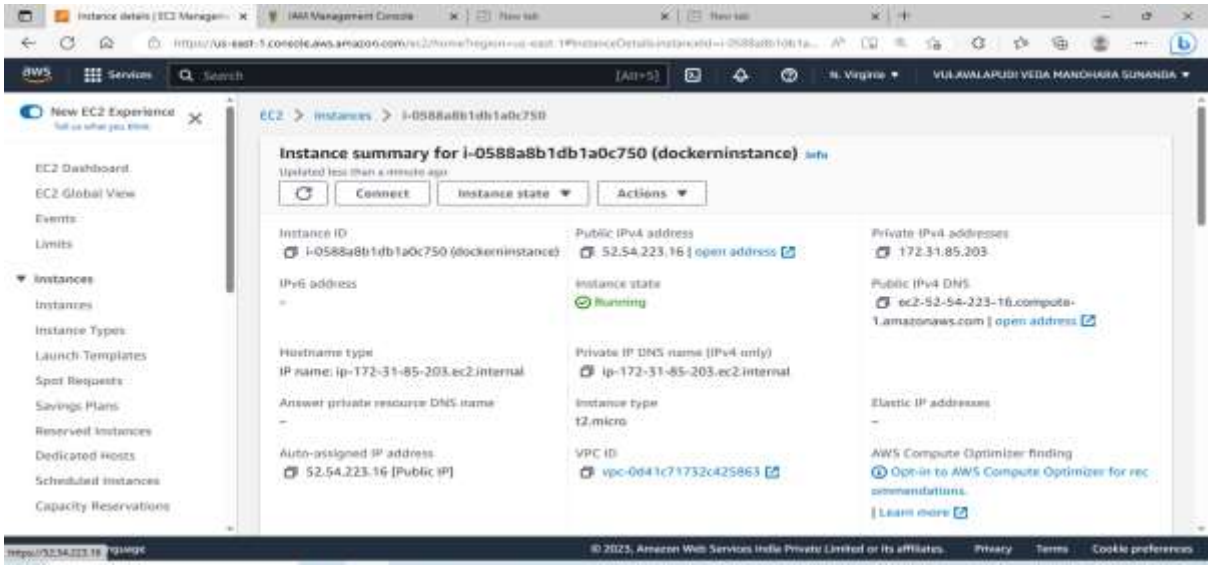
For GIT CLONE for ec2 automation EC2 Instances for computing
created by terraform

Docker Installed on EC2 Instance

Creating IAM user for Terraform automation







step1: create ec2 instance with linux ami

step2: use these commands

**Update the installed packages and package cache on your instance.

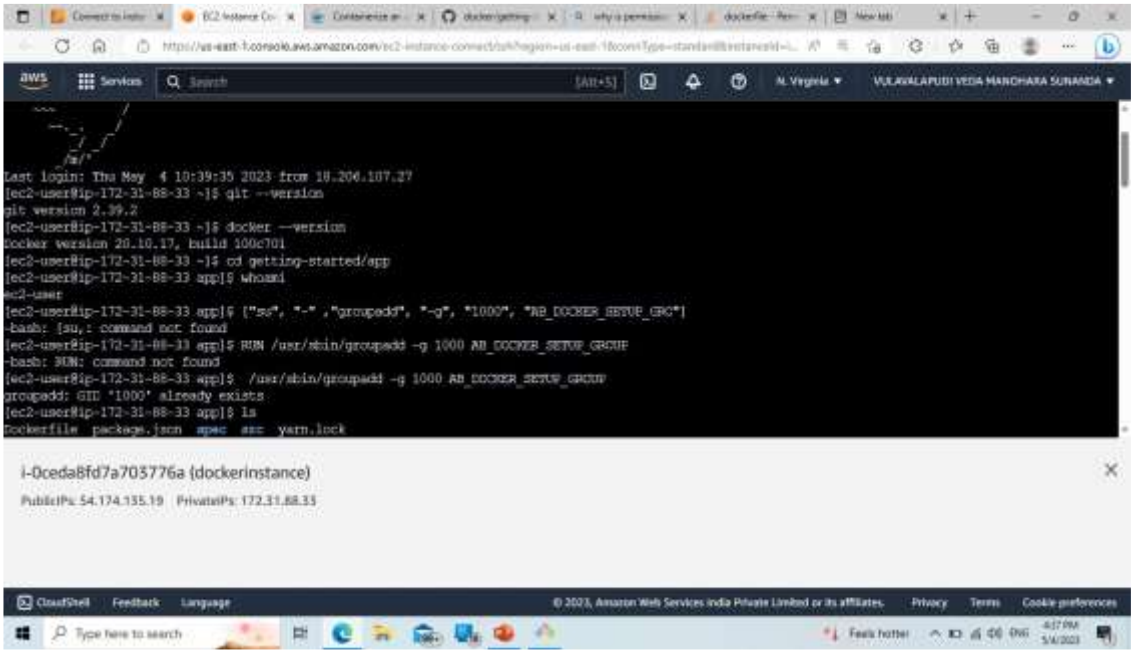
sudo yum update -y

**Install the most recent Docker Community Edition package.

sudo yum install docker

**Check docker version docker version

docker --version



**Add the ec2-user to the docker group so you can execute Docker commands without using sudo.

**Exit the terminal and re-login to make the change effective

```
sudo usermod -a -G docker ec2-user
exit
```

**Enable docker service

```
sudo systemctl enable docker
```

**Start docker service

```
sudo systemctl start docker
```

Check the Docker service.

```
sudo systemctl status docker
```

**Get docker details

docker info

**Get docker command details

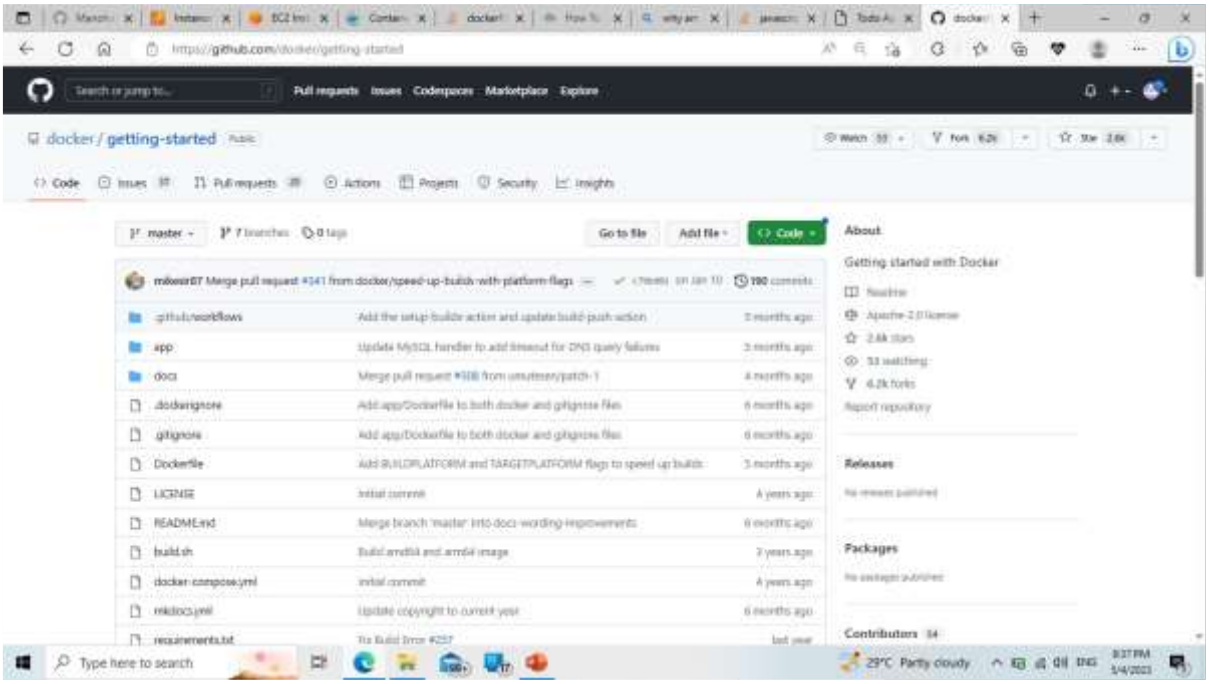
docker

**Get help on specific command

docker --help

**Installing Git

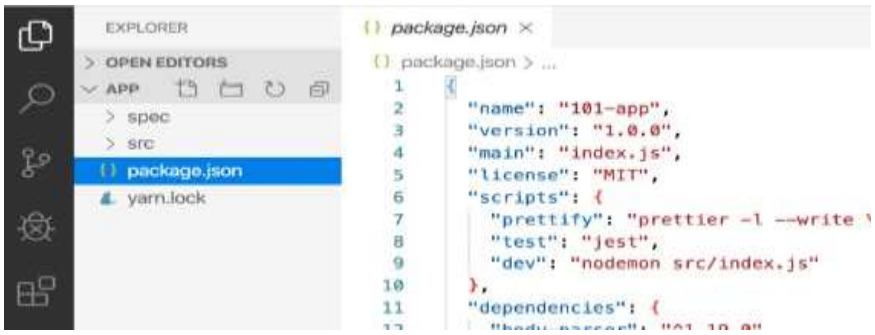
sudo yum install git

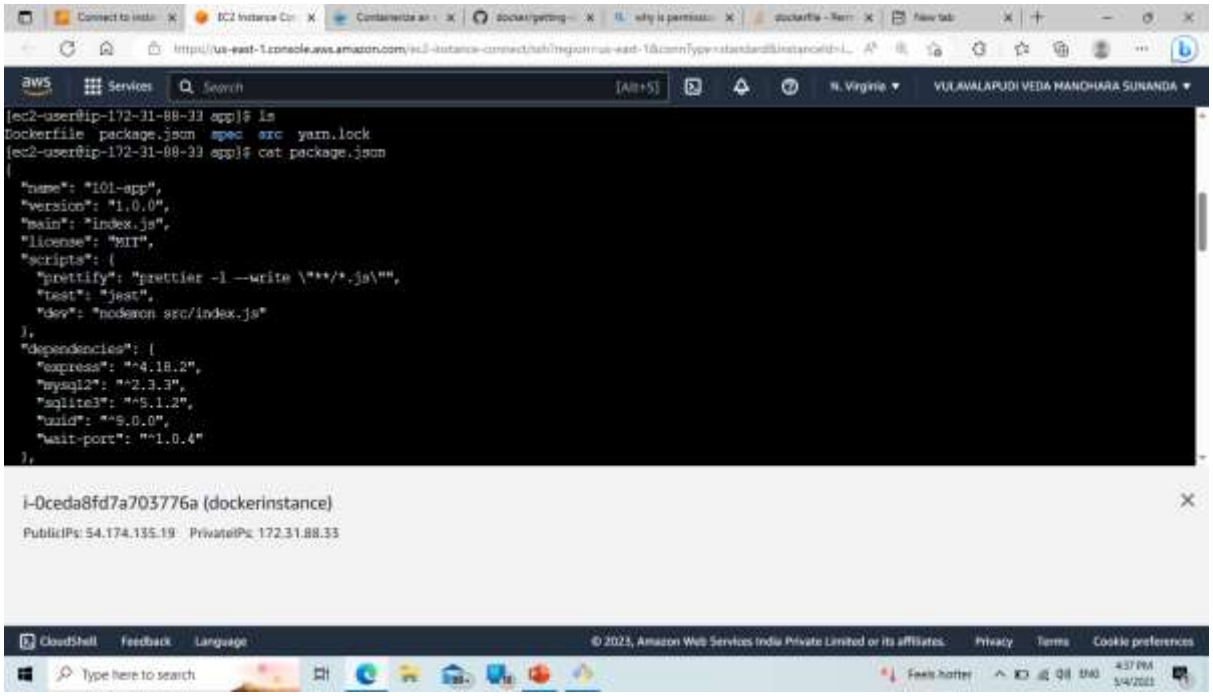


Working on project

git clone https://github.com/docker/getting-started.git

as below settings





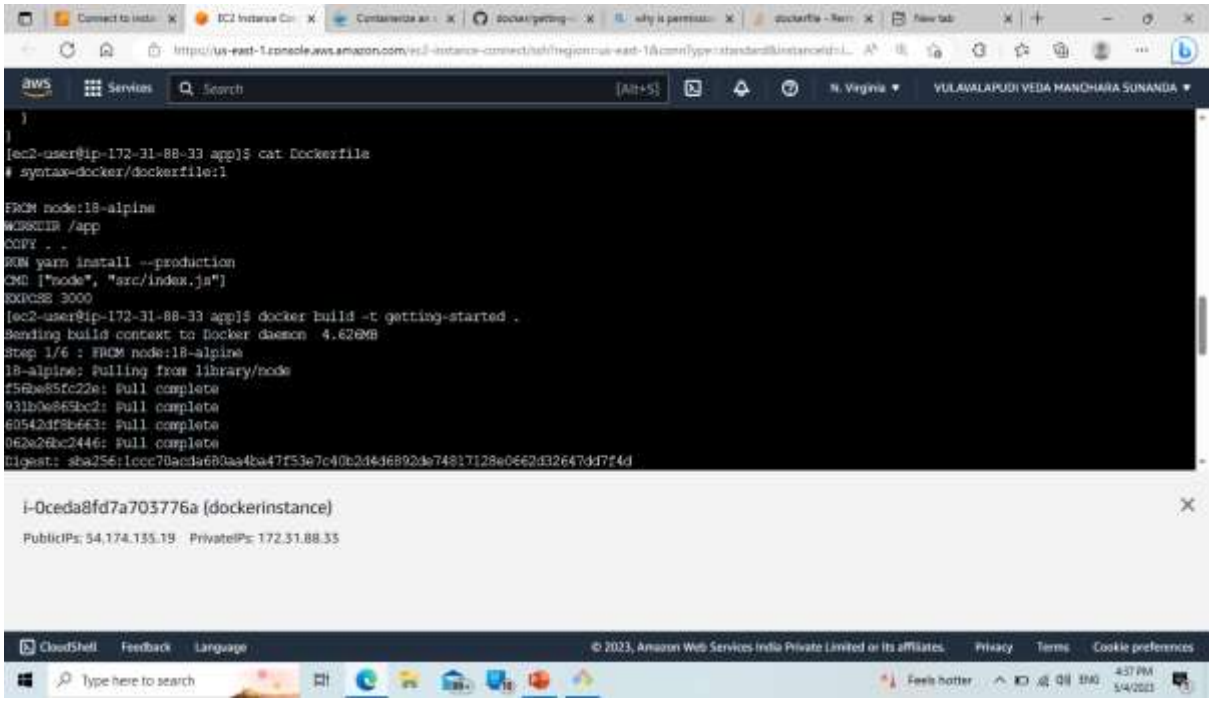
cd /path/to/app [cd getting-started/app]

touch Dockerfile

nano Dockerfile

syntax=docker/dockerfile:1

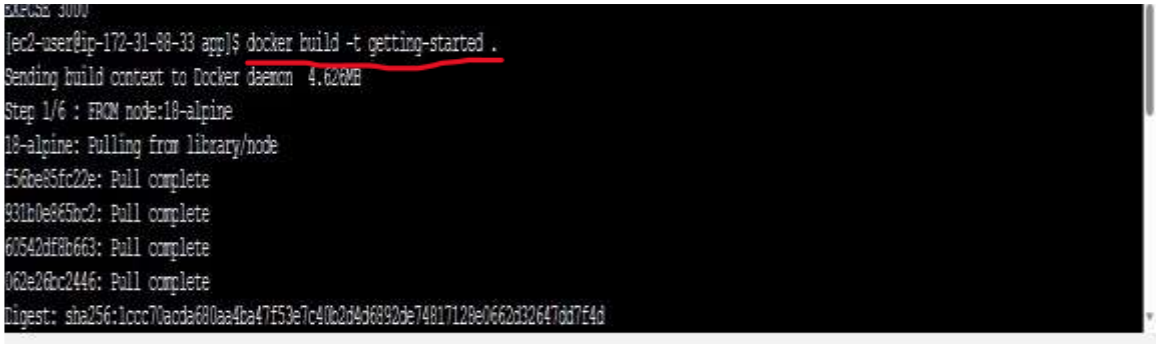
```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"] -->> to run index.js file
EXPOSE 3000   -> at port number 3000
```

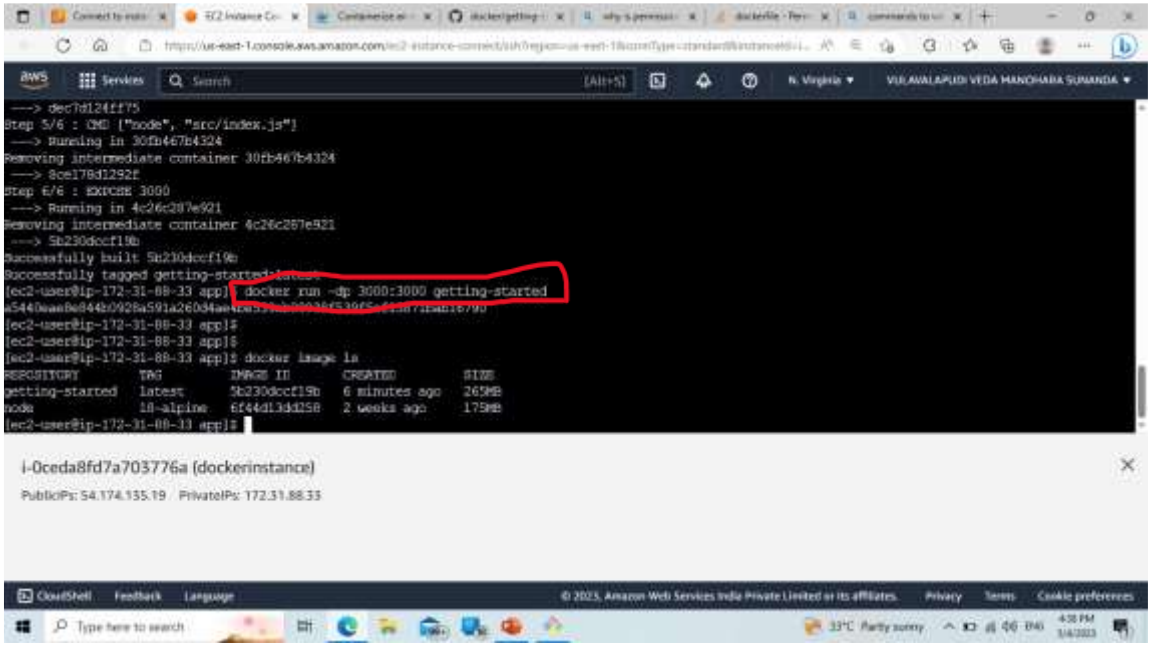


If "build" unsuccessful then use command

/usr/sbin/groupadd -g 1000 AB_DOCKER_SETUP_GROUP

and continue to work on same "build" command

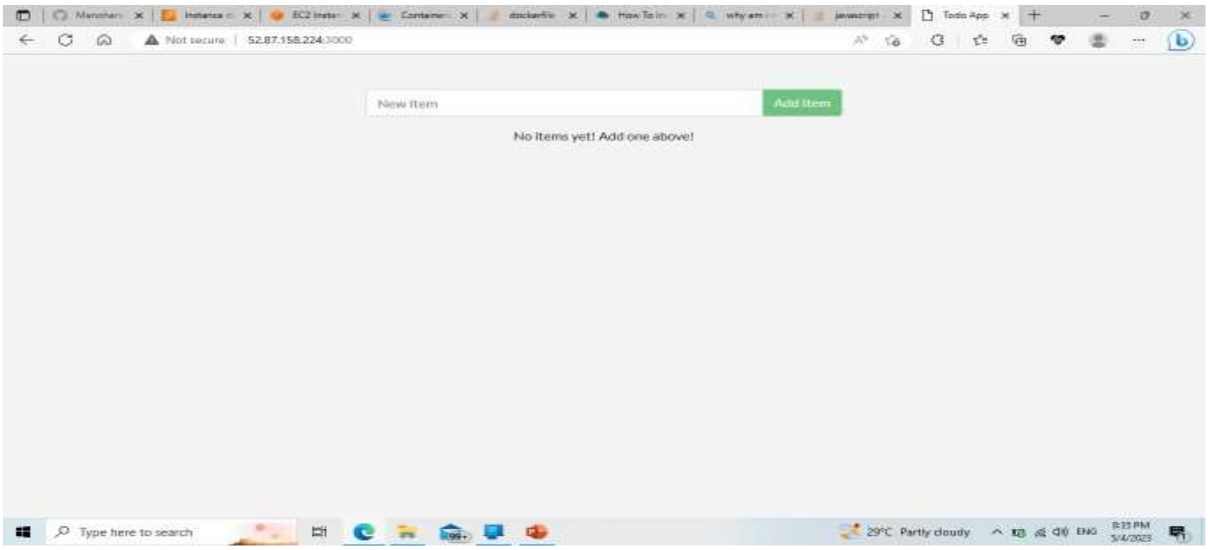




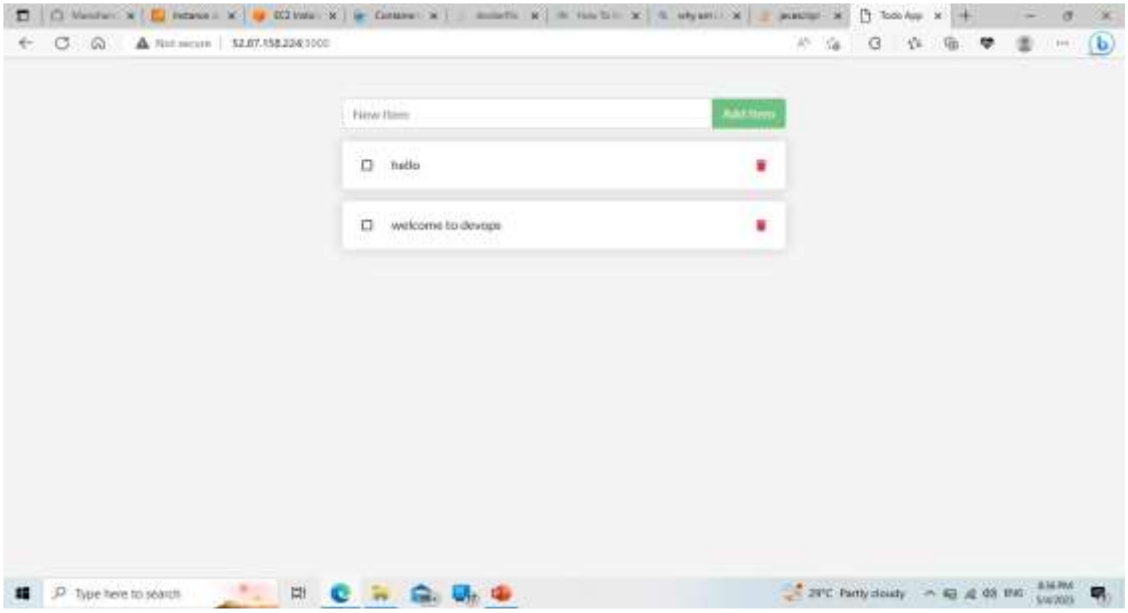
Work with the tags

docker run-dp 3000:3000 gettingstarted

The IP Addresses of the may be varied as I have terminated an instance due to billing issue and created the same in other instance with same procedure



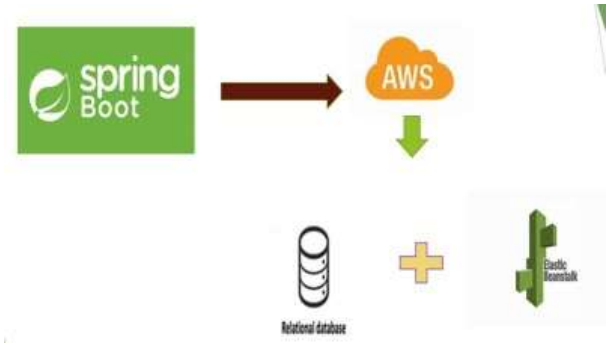
Dynamic addition of Items in “POST METHOD”



Project Number 2

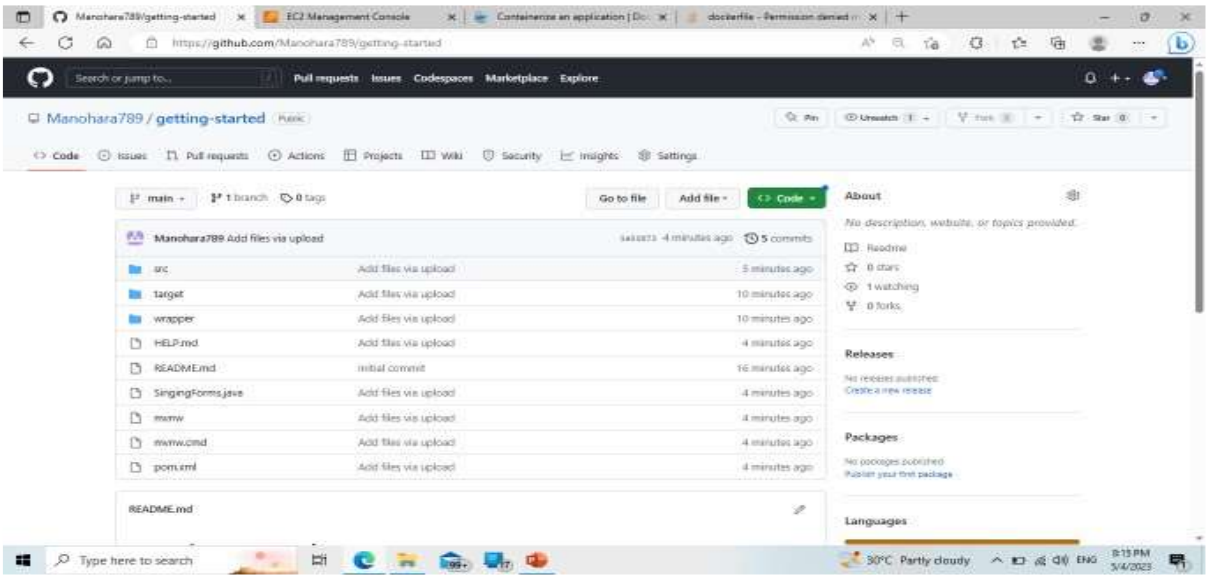
Deploying Spring boot Web Application , Automation via AWS Cloud

Services used are Git bash to push my files to my Git account +
Elastic Beanstalk +
RDS for configuring Application backend



- Step1 : create and SD Project of your own choice.
  We have created Indian Culture Information System Project
  Tools we used are : Spring boot, JavaScript for frontend
  Hibernate, JPA Mapping, MySQL for backend

This is my git with spring boot so you can also work on this in the same way

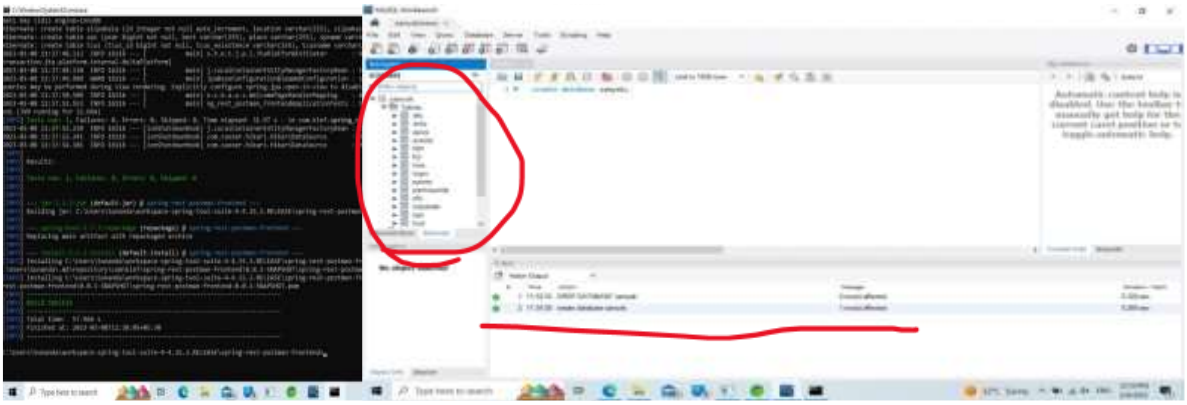


Output in the localhost machine port 8080

Steps I have followed to automate my application via EBS instance

Step1 : Create an RDS with required username, password and using its endpoint create a connection

Step2 : After establishing connection you copy endpoint to your "project application.properties" and open command prompt from the project location and enter "mvn clean install" to transfer tables to RDS database



Step 3 : Apply "mvn clean package" to get a snapshot of your project and then upload it to your EBS by selecting JAVA as your runtime and

Edit : 1. Environment variables PORT_NUMBER : 5000

2. Database Settings USERNAME : root

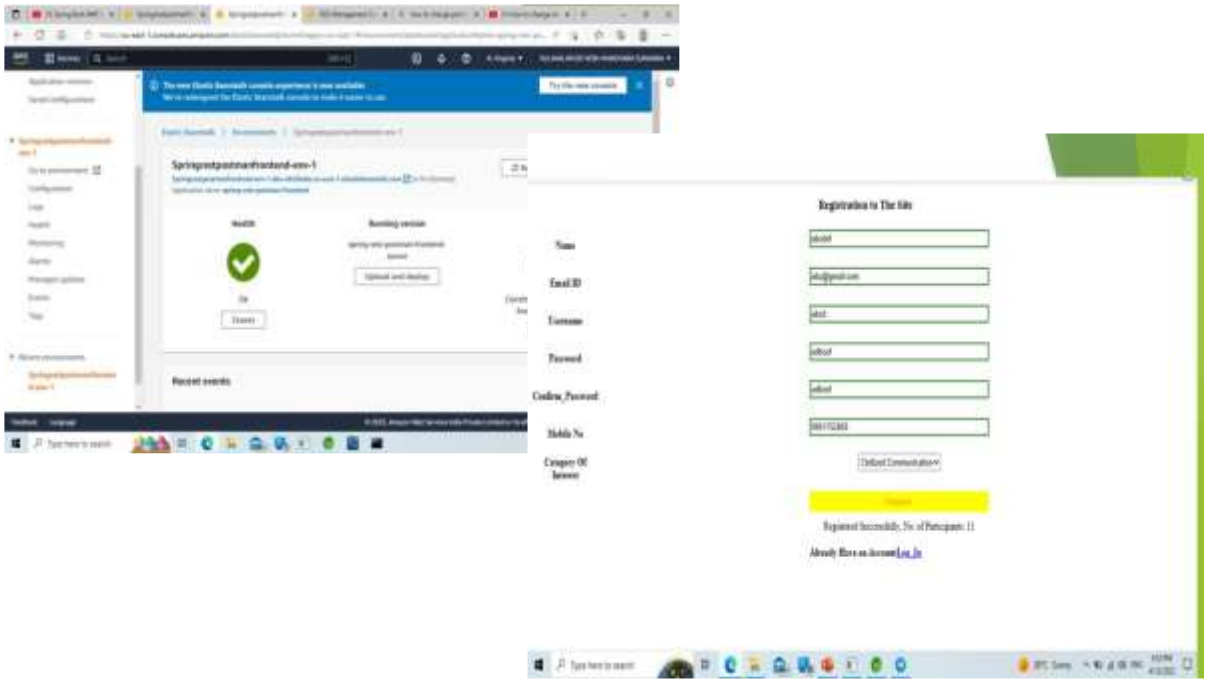
PASSWORD

Step 4 : Edit Auto-created EC2 Instance Security group inbound rules to allow backend data also

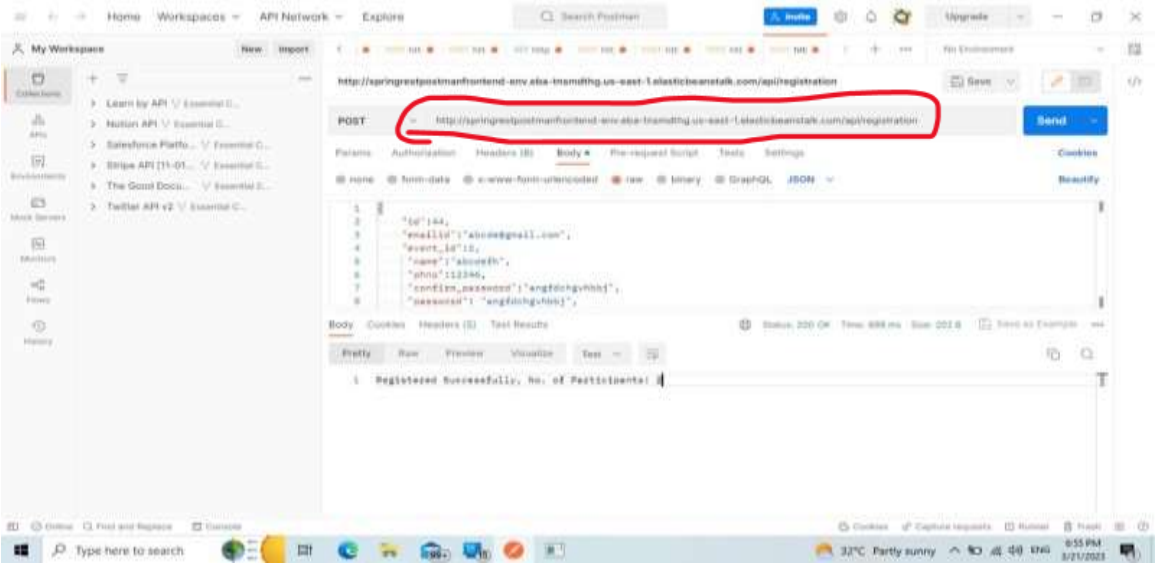
Step 5 : Take the help of EBS Application -> Environment URL and paste it in the POST MAN to check the request is working or not

- Step 6 : We tried the registration page URL "POST MAPPING" and finally it worked.

- We then analyzed that it has been hosted successfully.



Post mapping into the Registration page been successful. Observe the URL in POSTMAN App



Conclusion :

I have done 2 projects

1. Using Devops Services like – Git, Terraform, AWS Client Manageable Compute Services, Docker on EC2
2. Using AWS Cloud Services like – Elastic Beanstalk, RDS [MySQL]

Observations :

These both belongs to Deployment of Dynamic Web Application but they differ in Client-Side Automation.

This Client-Side Automation can be possible using services like Terraform, Docker Orchestration etc…

So, to show this difference I have done 2, 3-Tier Architecture Projects.

THANK YOU