Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

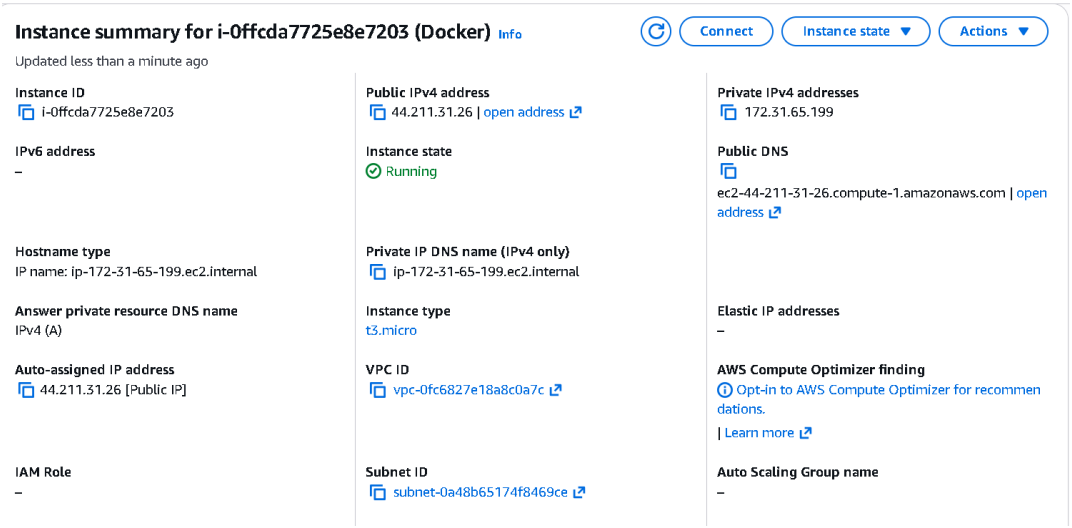
Title of Practicle : 3. Containerizing applications with Docker and Docker Compose

1. **Launch the instance.**

Name: Docker

AMI: Amazon Linux2023

Instance Type: t3.micro



1. **Install the Docker & Docker Compose.**

sudo yum update -y

sudo dnf install docker -y

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker ec2-user

sudo usermod -aG docker $USER

sudo curl -L "https://github.com/docker/compose/releases/download/v2.29.2/docker-compose-linux-x86_64" -o /usr/local/bin/docker-compose

Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

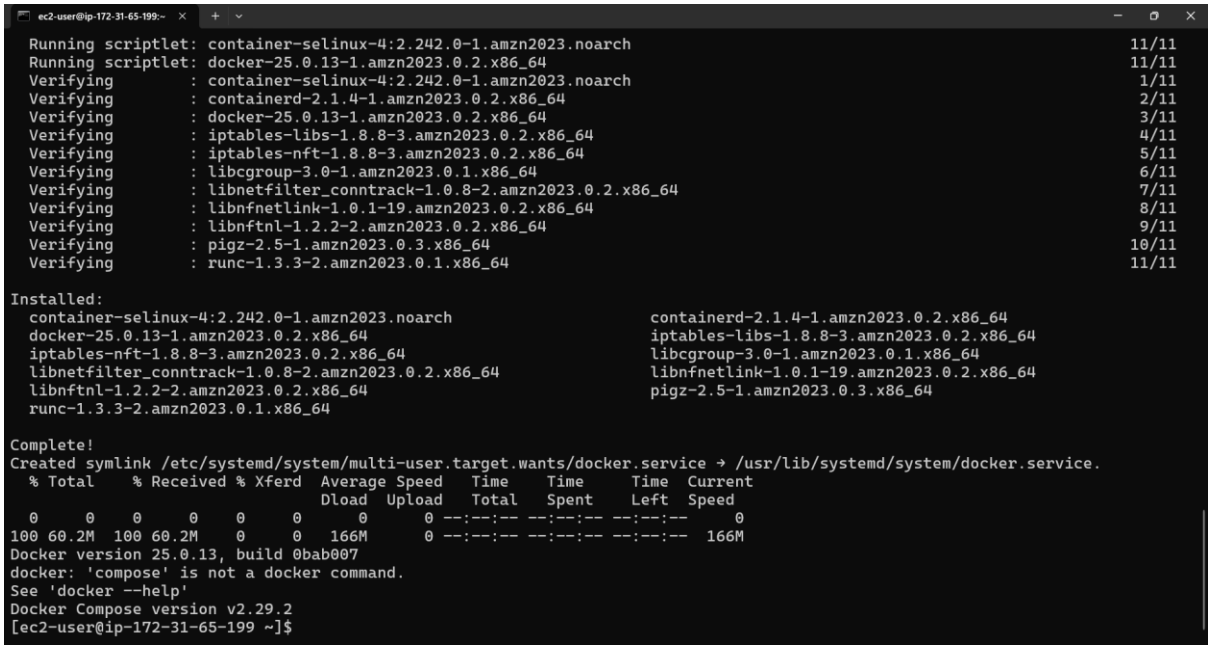
Title of Practicle : 3. Containerizing applications with Docker and Docker Compose

sudo chmod +x /usr/local/bin/docker-compose

docker --version

docker compose version

docker-compose version



Name of the Student:	**Pratik.M.Rebari**	**PRN:**	**20220802183**
Title of Practicle :	**3. Containerizing applications with Docker and Docker Compose**		

2. Create Your Project Folder

mkdir myapp

cd myapp

3. Add Your Application Code

Example: Node.js Express app

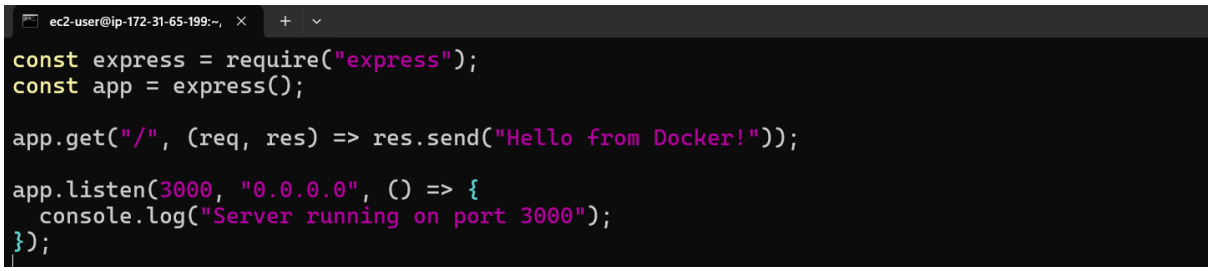
Create app.js

$ vi app.js

```
const express = require("express");

const app = express();


app.get("/", (req, res) => res.send("Hello from Docker!"));


app.listen(3000, () => console.log("Server running on port 3000"));
```

Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

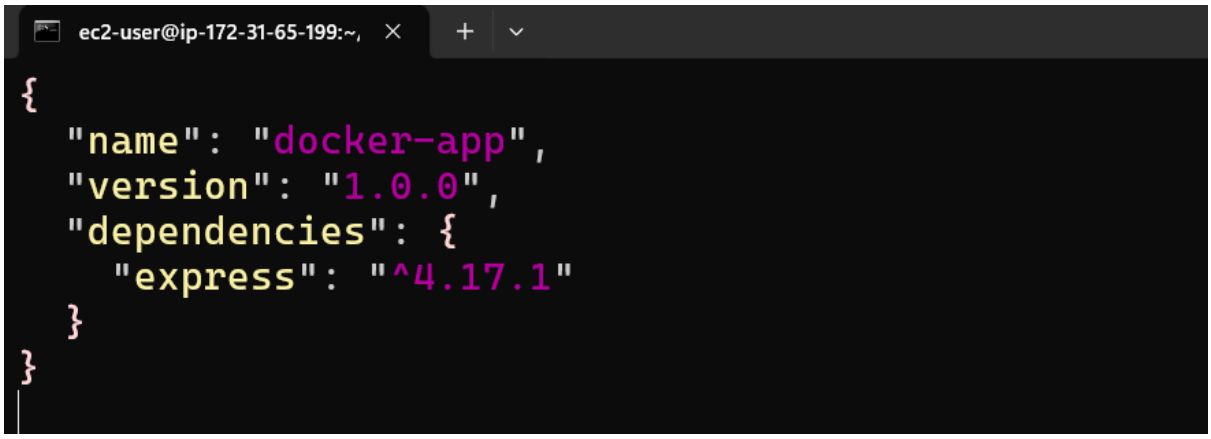
Title of Practicle : 3. Containerizing applications with Docker and Docker Compose



Create package.json

$ package.json

```
{
 "name": "docker-app",
 "version": "1.0.0",
 "dependencies": {
  "express": "^4.17.1"
 }
}
```



Name of the Student:	**Pratik.M.Rebari**	**PRN:**	**20220802183**
Title of Practicle :	**3. Containerizing applications with Docker and Docker Compose**		

4. Write the Dockerfile

Create Dockerfile

$ vi dockerfile

FROM node:18

WORKDIR /app

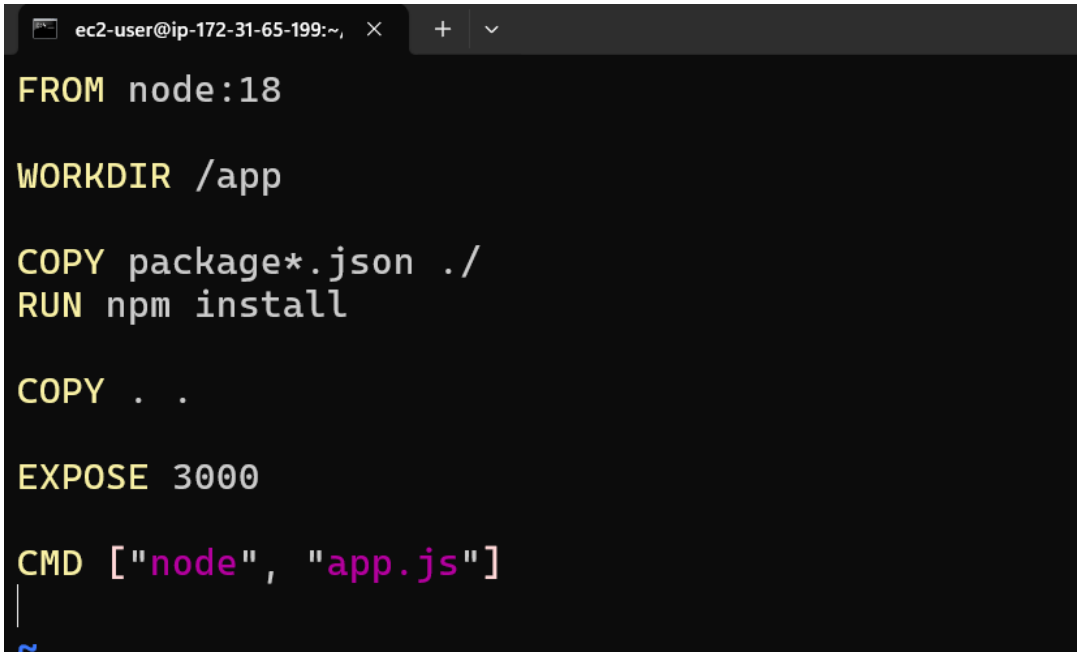
COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "app.js"]

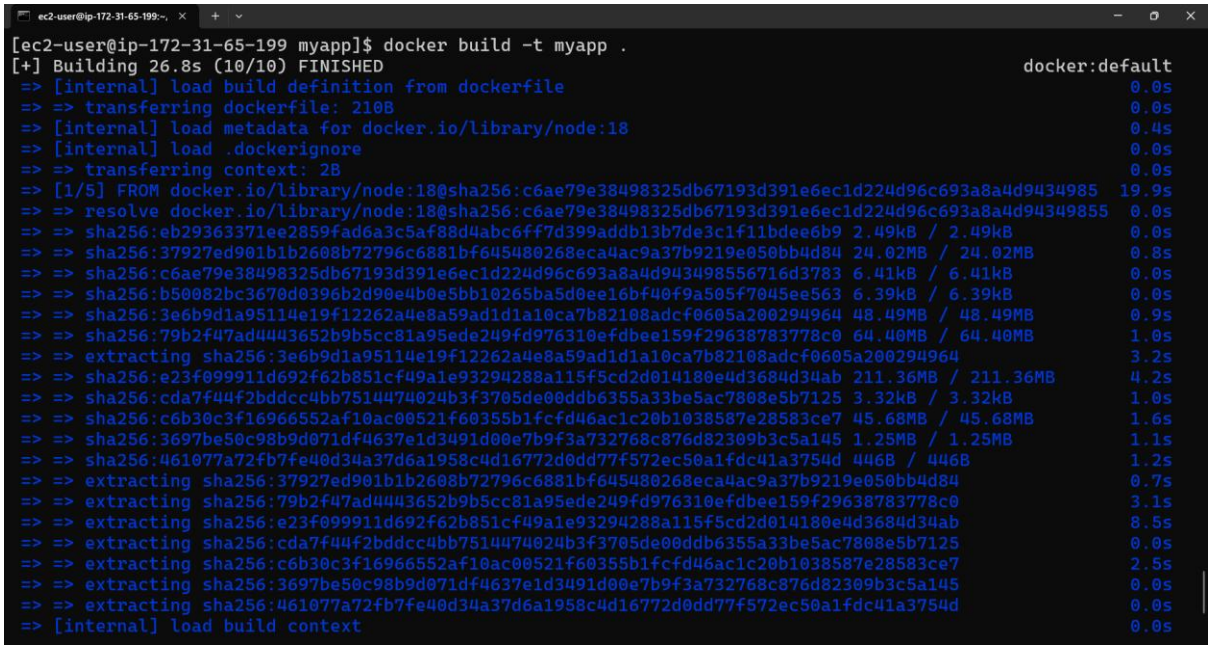


Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

Title of Practicle : 3. Containerizing applications with Docker and Docker Compose

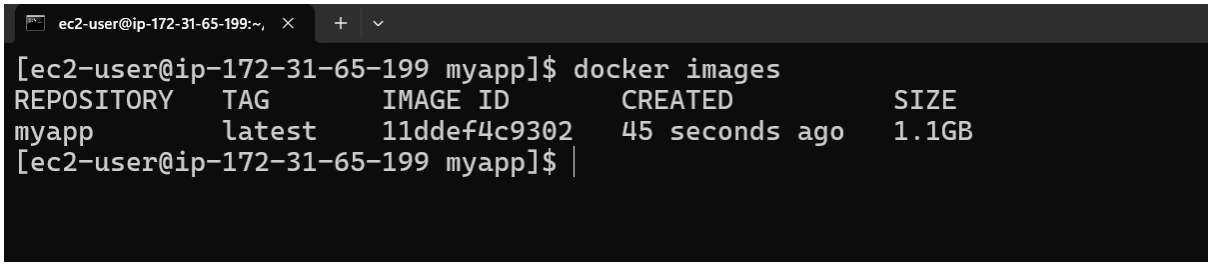
5. Build Docker Image

$ docker build -t myapp .



Check image:

$ docker images

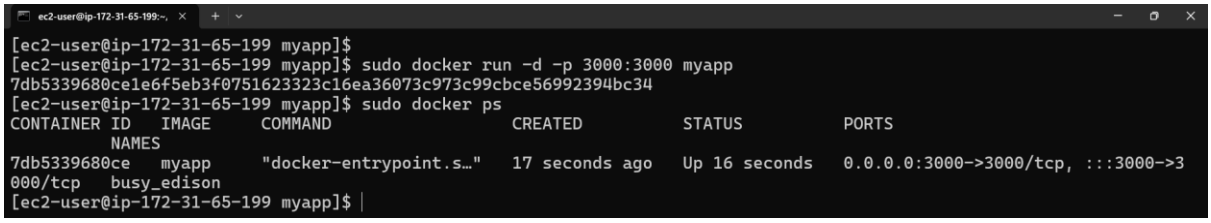


Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

Title of Practicle : 3. Containerizing applications with Docker and Docker Compose

6. Run Container (Without Compose)

$ docker run -d -p 3000:3000 --name myapp_container myapp



Test:

http://<ip>:3000

Hello from Docker!

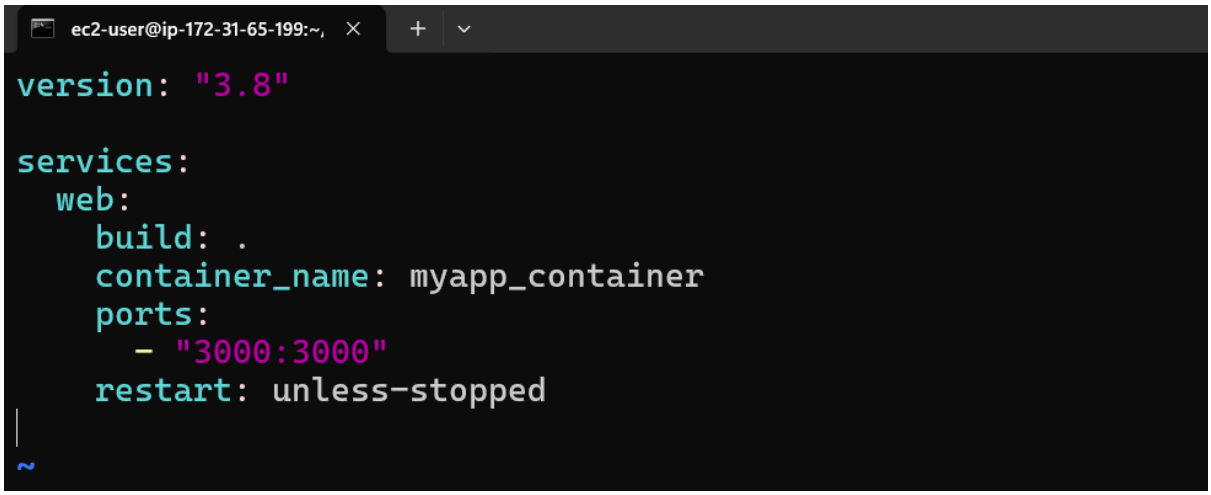
7. Create docker-compose.yml

$ docker-compose.yml

```
version: "3.8"
services:
  web:
    build: .
    container_name: myapp_container
    ports:
      - "3000:3000"
    restart: unless-stopped
```

Name of the Student: Pratik.M.Rebari **PRN:** 20220802183

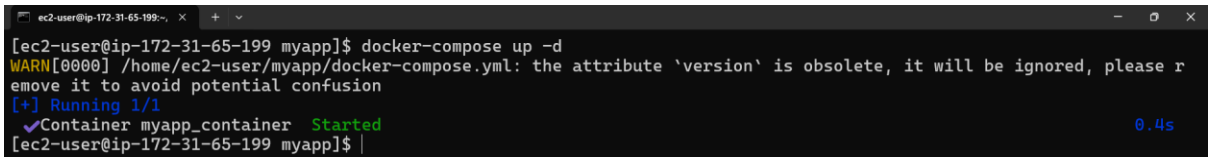
Title of Practicle : 3. Containerizing applications with Docker and Docker Compose



8. Run Using Docker Compose

Start:

$ docker-compose up -d



Check running:

$ docker ps