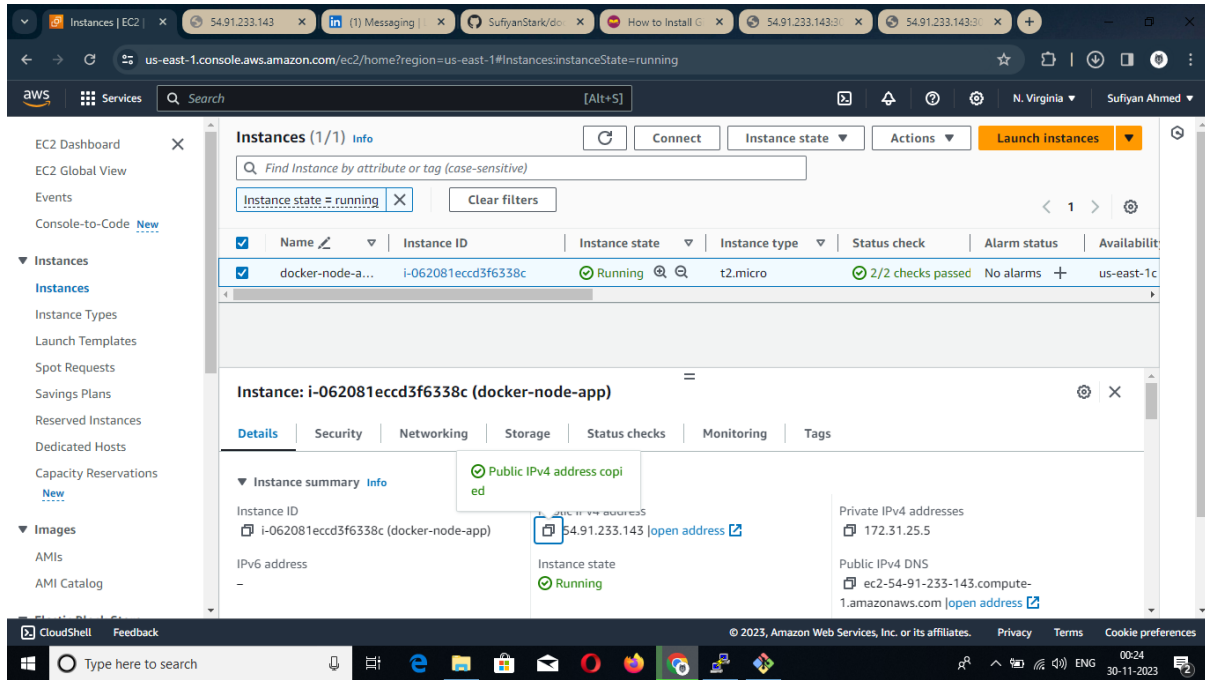


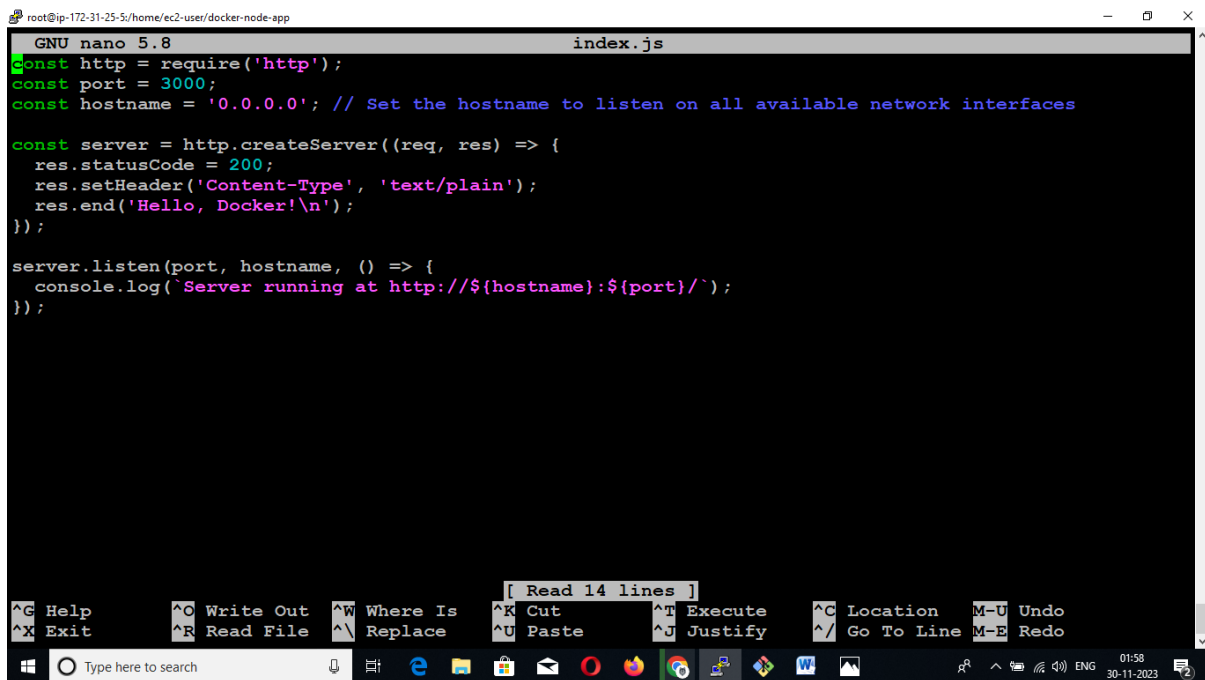
# Assessment

I have done the assessment using AWS ill be putting the screenshots below:-

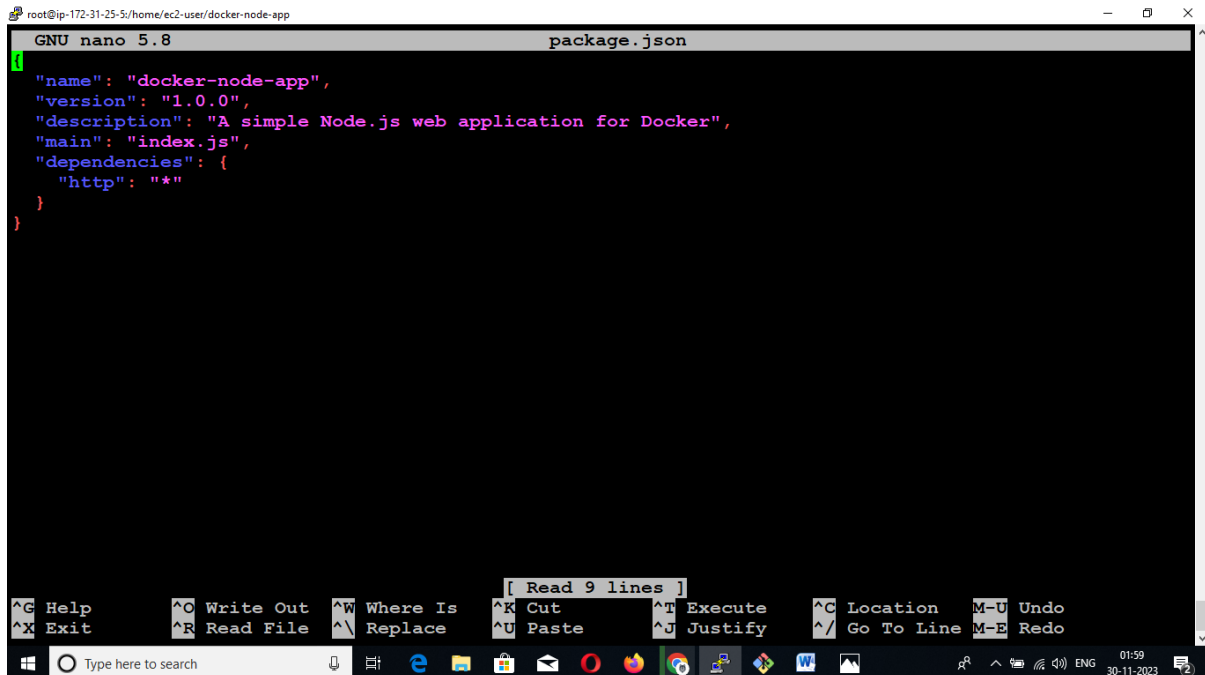


I made two file that is index.js and package.json and pushed it to my github

index.js



## Package.json:-

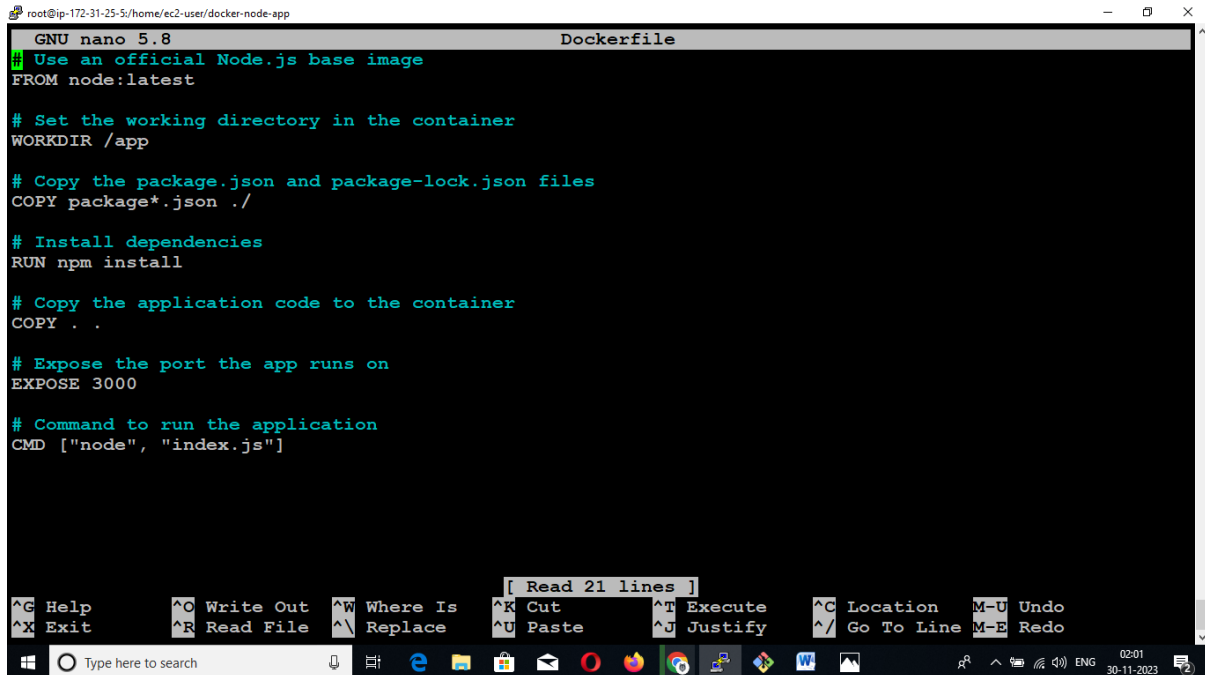


A screenshot of a terminal window titled "root@ip-172-31-25-5:/home/ec2-user/docker-node-app". The terminal shows the GNU nano 5.8 editor editing a file named "package.json". The content of the file is a JSON object with the following fields: "name" (docker-node-app), "version" (1.0.0), "description" (A simple Node.js web application for Docker), "main" (index.js), and "dependencies" (http: \*). The terminal window has a Windows taskbar at the bottom with various application icons and a search bar.

```
GNU nano 5.8 package.json
{
  "name": "docker-node-app",
  "version": "1.0.0",
  "description": "A simple Node.js web application for Docker",
  "main": "index.js",
  "dependencies": {
    "http": "*"
  }
}
```

then I cloned it on my aws instance as you can see in the above screenshots

after cloning In the docker-node-app directory, I created a file named Dockerfile (without any file extension) using a text editor and add the following content



A screenshot of a terminal window titled "root@ip-172-31-25-5:/home/ec2-user/docker-node-app". The terminal shows the GNU nano 5.8 editor editing a file named "Dockerfile". The content of the file is a Dockerfile with the following instructions: "FROM node:latest", "WORKDIR /app", "COPY package\*.json ./", "RUN npm install", "COPY . .", "EXPOSE 3000", and "CMD [\"node\", \"index.js\"]". The terminal window has a Windows taskbar at the bottom with various application icons and a search bar.

```
GNU nano 5.8 Dockerfile
# Use an official Node.js base image
FROM node:latest

# Set the working directory in the container
WORKDIR /app

# Copy the package.json and package-lock.json files
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the application code to the container
COPY . .

# Expose the port the app runs on
EXPOSE 3000

# Command to run the application
CMD ["node", "index.js"]
```

Build the Docker image using the following command:

```
docker build -t my-node-app .
```

Now to make user running the docker build command has the necessary permissions:

```
sudo docker build -t my-node-app-image .
```

confirm the successful creation of the Docker image by running the following command:

```
sudo docker images
```

To run a Docker container from the newly created image, use the docker run command:

```
sudo docker run -d -p 3000:3000 my-node-app-image
```

After running the container, your Node.js application should be accessible at

<http://54.91.233.143:3000>

the output I got at the last as successful running of the container and access to node.js application :-

