# Experiment No : 5

Title: Running Node.js application over docker.

Aim : To run node.js application over docker.

Theory:

**What is Docker?**

Docker is a software platform. It enables software developers to develop, ship and run applications within its containers. Containers are lightweight software applications. We are going to build a Docker image in this experiment.

**What is a docker file, image and container?**

A docker file is a text file that contains the set of instructions for the Docker platform. Therefore, it can be versioned and committed to a code repository.

An image includes everything needed to run an application — the code or binary, runtime, dependencies, and any other file system objects required.

Docker containers run the application code.

we will create a simple web application in Node.js, then we will build a Docker image for that application, and lastly we will instantiate a container from that image.

Docker allows you to package an application with its environment and all of its dependencies into a "box", called a container. Usually, a container consists of an application running in a stripped-to-basics version of a Linux operating system. An image is the blueprint for a container, a container is a running instance of an image.

What Is Node.js?

Docker Node.js or simply, Node.js is an open-source software platform used to build scalable server-side and network applications. These Node.js applications are written in JavaScript and can run within this Node.js runtime on Linux, Windows, or Mac OS without changes. It was originally designed keeping in mind real-time and push-based architecture. Nowadays, it is primarily used for non-blocking, event-driven servers like backend API services and traditional websites.

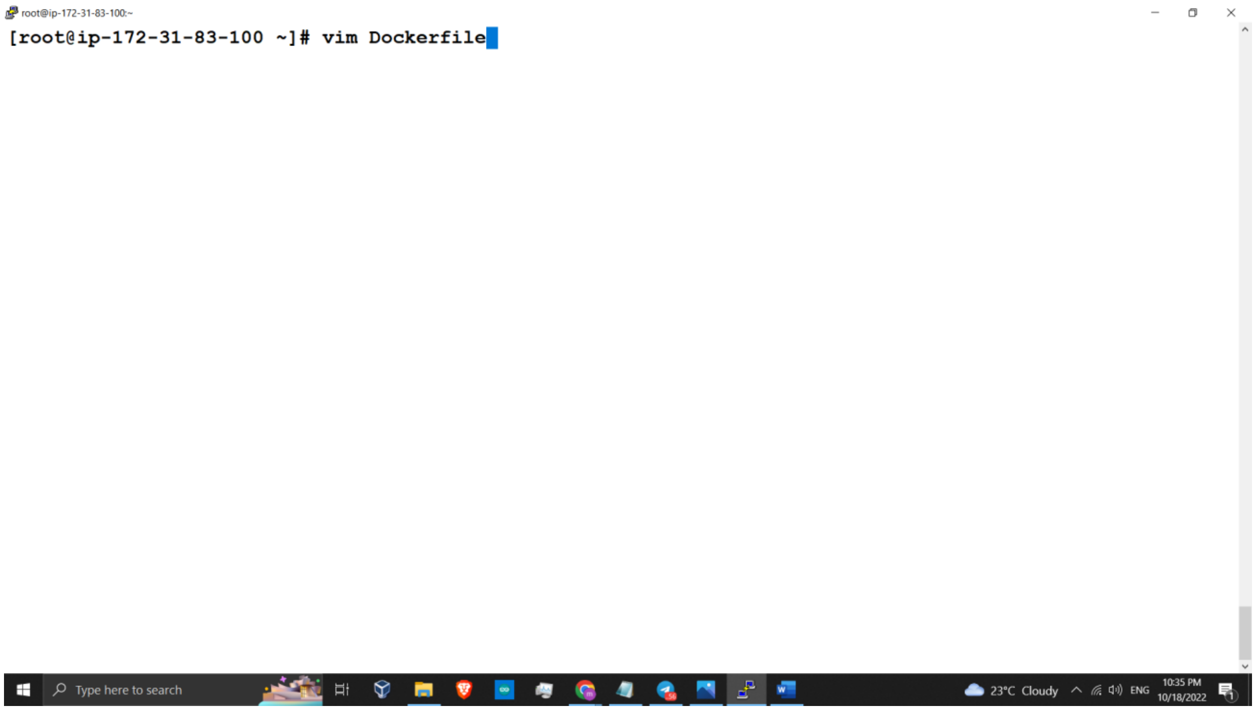
Docker Node.js uses the Google V8 JavaScript to execute code and has its own built-in library for file, socket, and HTTP communication. This allows Node.js to act as a web server on its own without the support of additional software.

Prerequisites

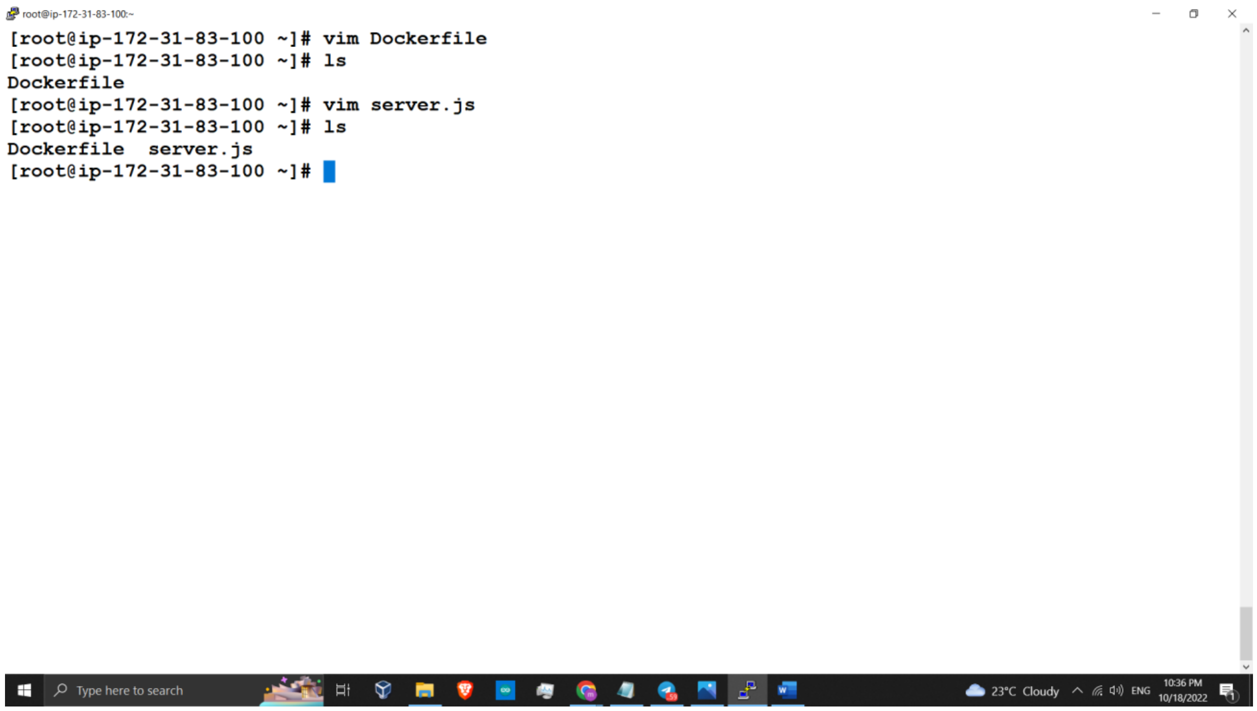
To build an application, you need to have the following things in place:

- Node.js version 12.18 or later
- Docker running locally
- A text editor or IDE to edit your files

**Practical:**

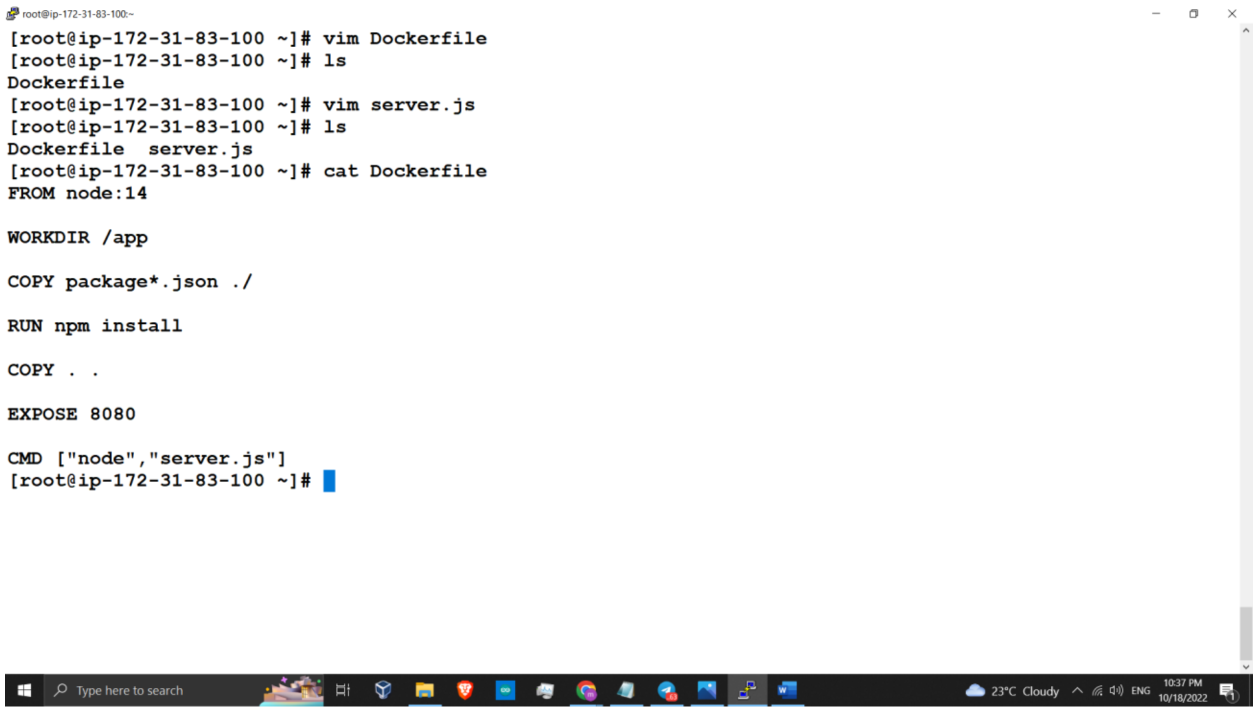


```
[root@ip-172-31-83-100 ~]# vim Dockerfile
```

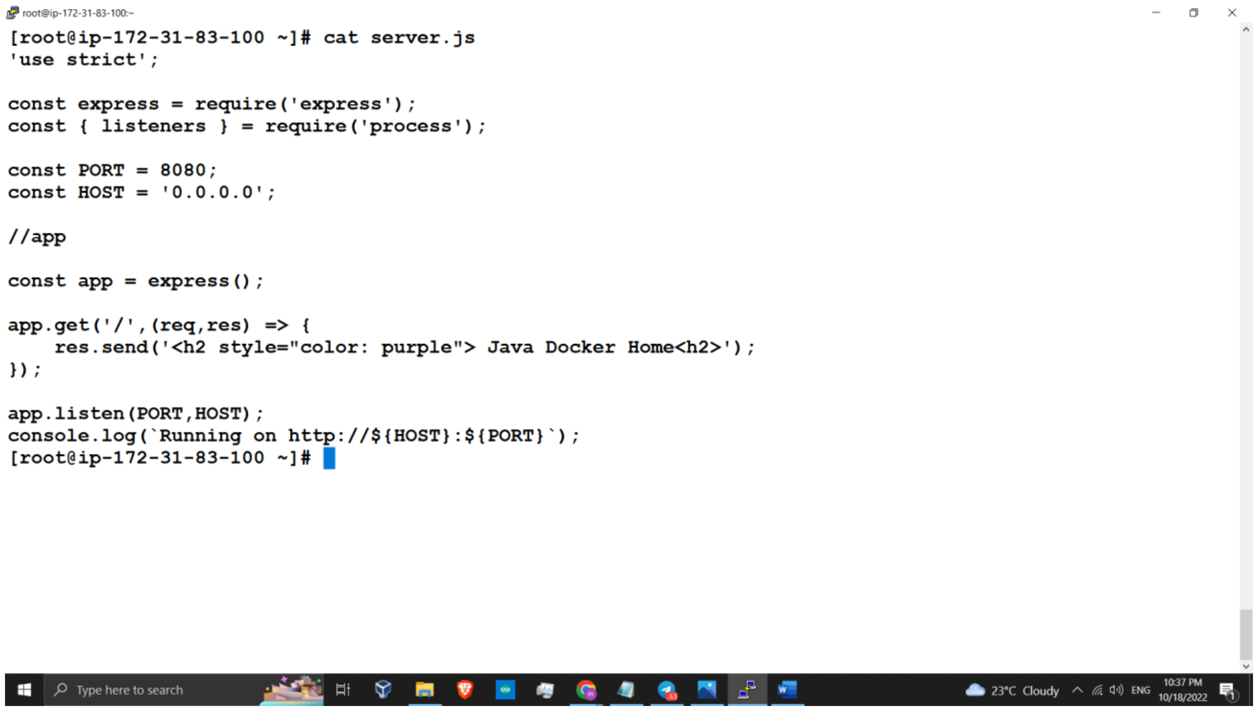


root@ip-172-31-83-100:~

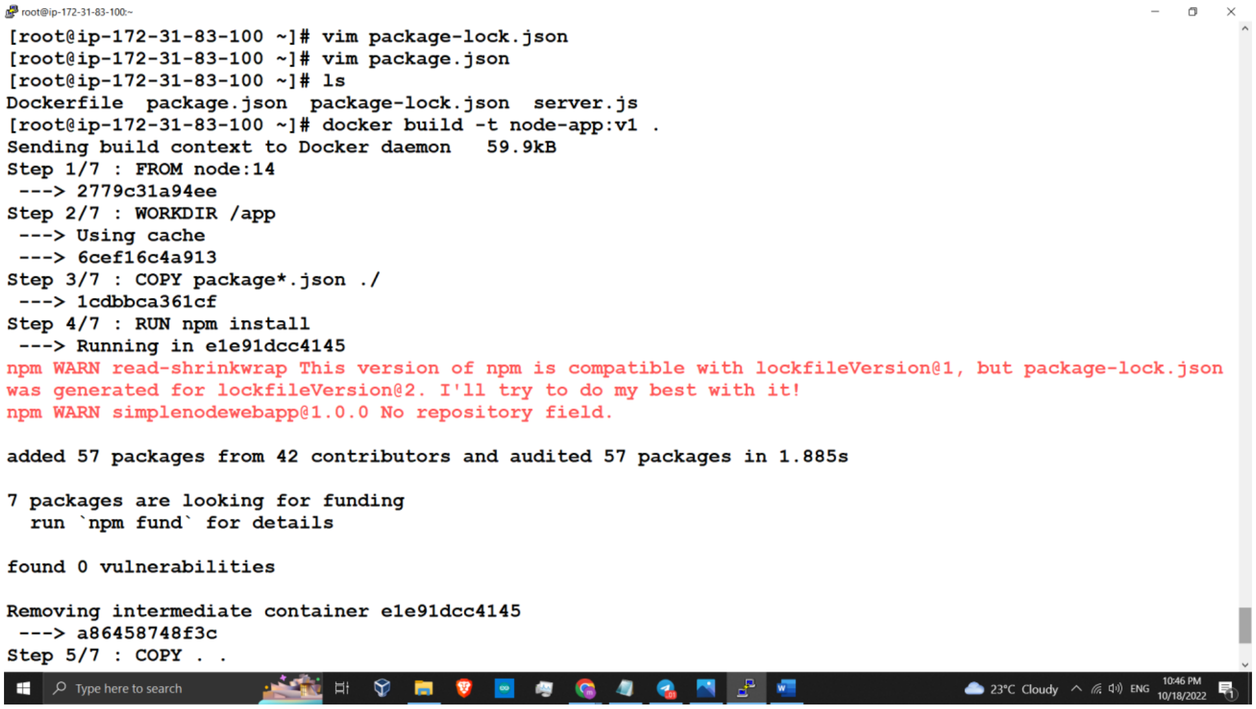
```
[root@ip-172-31-83-100 ~]# vim Dockerfile
[root@ip-172-31-83-100 ~]# ls
Dockerfile
[root@ip-172-31-83-100 ~]# vim server.js
[root@ip-172-31-83-100 ~]# ls
Dockerfile  server.js
[root@ip-172-31-83-100 ~]#
```



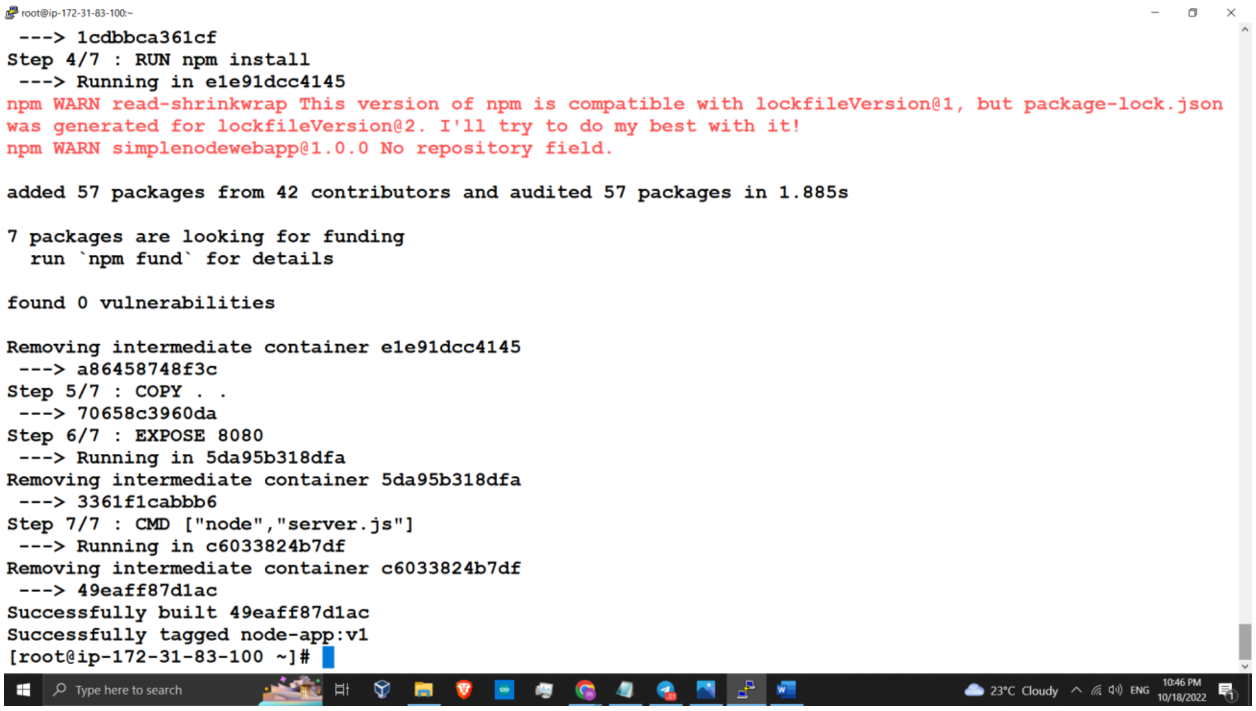
```
[root@ip-172-31-83-100 ~]# vim Dockerfile
[root@ip-172-31-83-100 ~]# ls
Dockerfile
[root@ip-172-31-83-100 ~]# vim server.js
[root@ip-172-31-83-100 ~]# ls
Dockerfile  server.js
[root@ip-172-31-83-100 ~]# cat Dockerfile
FROM node:14

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 8080

CMD ["node","server.js"]
[root@ip-172-31-83-100 ~]#
```



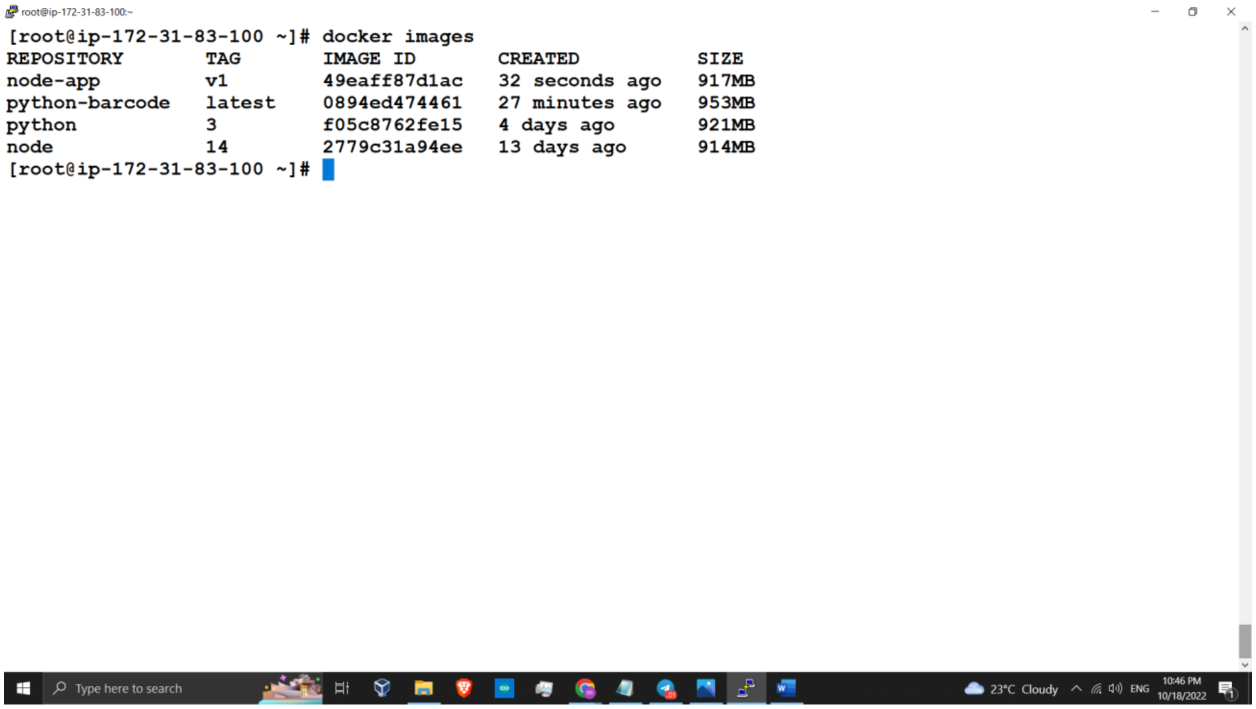
```
[root@ip-172-31-83-100 ~]# cat server.js
'use strict';

const express = require('express');
const { listeners } = require('process');

const PORT = 8080;
const HOST = '0.0.0.0';

//app

const app = express();

app.get('/',(req,res) => {
    res.send('<h2 style="color: purple"> Java Docker Home<h2>');
});

app.listen(PORT,HOST);
console.log(`Running on http://${HOST}:${PORT}`);
[root@ip-172-31-83-100 ~]#
```



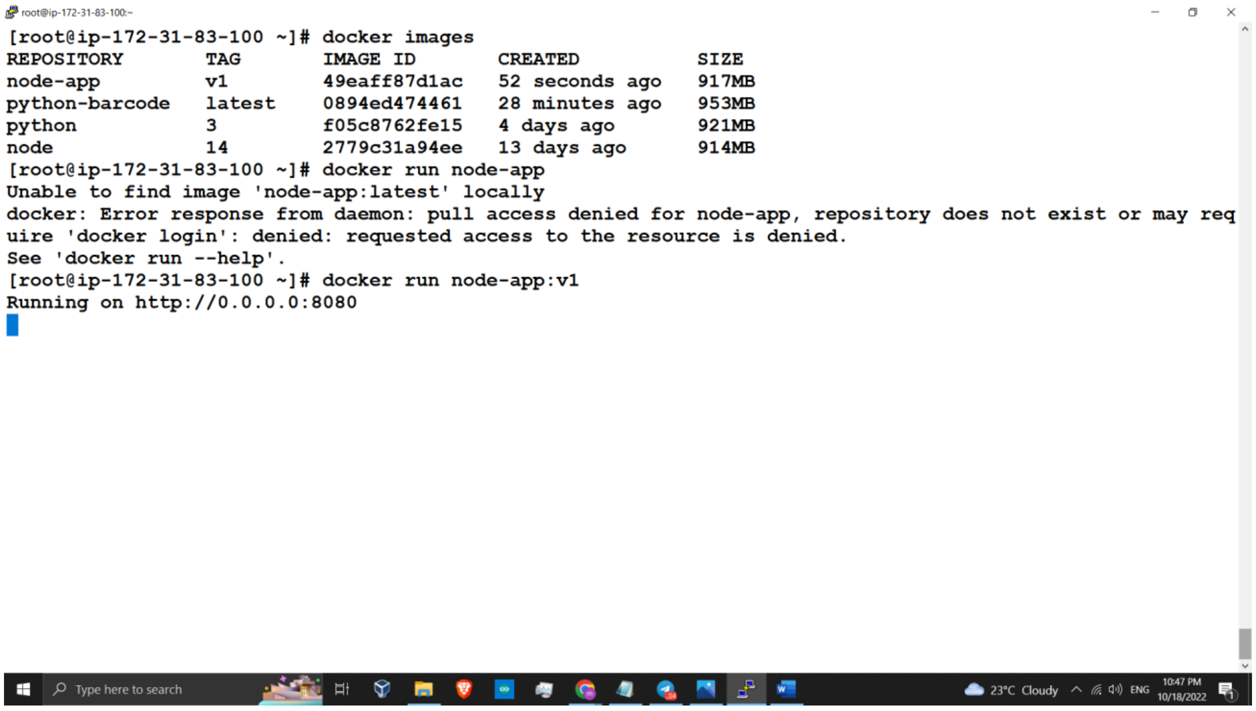
```
[root@ip-172-31-83-100 ~]# vim package-lock.json
[root@ip-172-31-83-100 ~]# vim package.json
[root@ip-172-31-83-100 ~]# ls
Dockerfile  package.json  package-lock.json  server.js
[root@ip-172-31-83-100 ~]# docker build -t node-app:v1 .
Sending build context to Docker daemon   59.9kB
Step 1/7 : FROM node:14
 ---> 2779c31a94ee
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 6cef16c4a913
Step 3/7 : COPY package*.json ./
 ---> 1cdbbca361cf
Step 4/7 : RUN npm install
 ---> Running in e1e91dcc4145
npm WARN read-shrinkwrap This version of npm is compatible with lockfileVersion@1, but package-lock.json
was generated for lockfileVersion@2. I'll try to do my best with it!
npm WARN simplenodewebapp@1.0.0 No repository field.

added 57 packages from 42 contributors and audited 57 packages in 1.885s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Removing intermediate container e1e91dcc4145
 ---> a86458748f3c
Step 5/7 : COPY . .
```



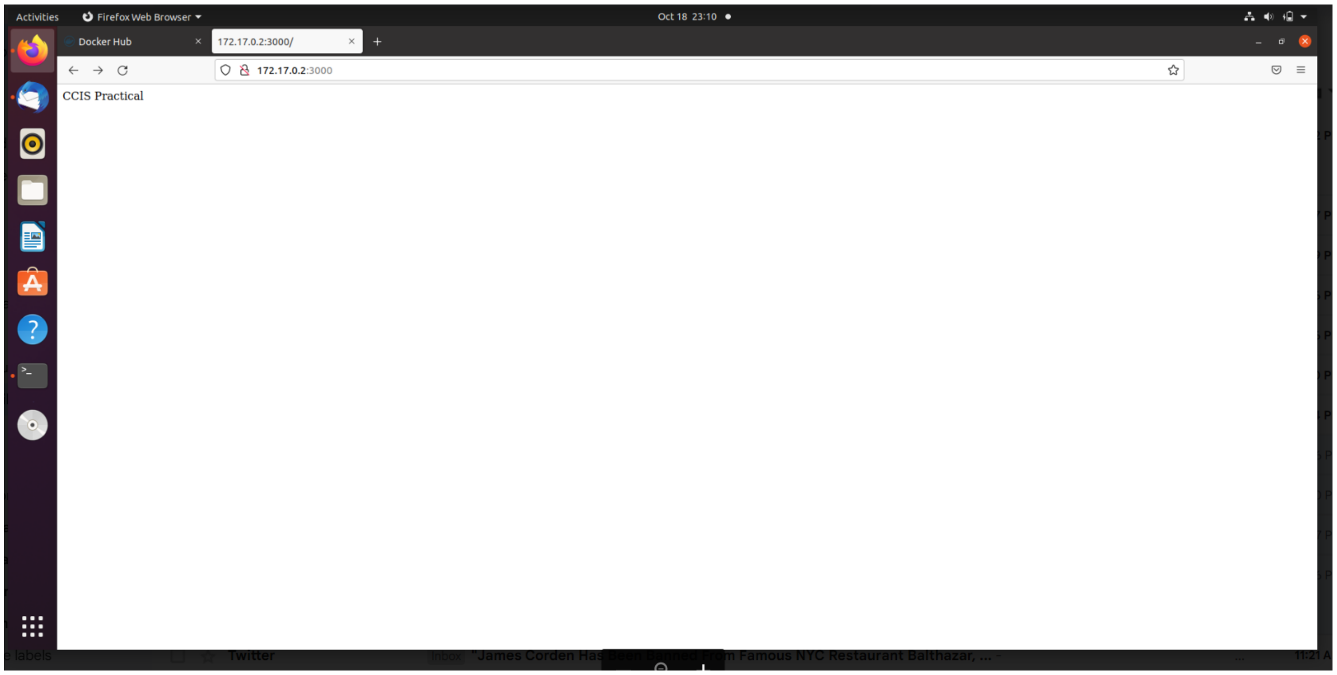
```
 ---> 1cdbbca361cf
Step 4/7 : RUN npm install
 ---> Running in e1e91dcc4145
npm WARN read-shrinkwrap This version of npm is compatible with lockfileVersion@1, but package-lock.json
was generated for lockfileVersion@2. I'll try to do my best with it!
npm WARN simplenodewebapp@1.0.0 No repository field.

added 57 packages from 42 contributors and audited 57 packages in 1.885s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Removing intermediate container e1e91dcc4145
 ---> a86458748f3c
Step 5/7 : COPY . .
 ---> 70658c3960da
Step 6/7 : EXPOSE 8080
 ---> Running in 5da95b318dfa
Removing intermediate container 5da95b318dfa
 ---> 3361f1cabbb6
Step 7/7 : CMD ["node","server.js"]
 ---> Running in c6033824b7df
Removing intermediate container c6033824b7df
 ---> 49eaff87d1ac
Successfully built 49eaff87d1ac
Successfully tagged node-app:v1
[root@ip-172-31-83-100 ~]#
```


root@ip-172-31-83-100:~

```
[root@ip-172-31-83-100 ~]# docker images
REPOSITORY       TAG       IMAGE ID       CREATED          SIZE
node-app         v1        49eaff87d1ac   32 seconds ago   917MB
python-barcode   latest    0894ed474461   27 minutes ago   953MB
python           3         f05c8762fe15   4 days ago       921MB
node             14        2779c31a94ee   13 days ago      914MB
[root@ip-172-31-83-100 ~]#
```


root@ip-172-31-83-100:~

```
[root@ip-172-31-83-100 ~]# docker images
REPOSITORY       TAG       IMAGE ID       CREATED          SIZE
node-app         v1        49eaff87d1ac   52 seconds ago   917MB
python-barcode   latest    0894ed474461   28 minutes ago   953MB
python           3         f05c8762fe15   4 days ago       921MB
node             14        2779c31a94ee   13 days ago      914MB
[root@ip-172-31-83-100 ~]# docker run node-app
Unable to find image 'node-app:latest' locally
docker: Error response from daemon: pull access denied for node-app, repository does not exist or may req
uire 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[root@ip-172-31-83-100 ~]# docker run node-app:v1
Running on http://0.0.0.0:8080
```



CCIS Practical

Conclusion: Thus we have run the node.js application on docker