

DevOps Assignment

Name: Hari Mahesh

Mobile: 8019782344

Email: harimahesh417@gmail.com

Objective: Demonstrate your basic ability to work with simple Docker features.

Deadline: 3 Days

Step 1: Download the assignment files from this link-

https://drive.google.com/file/d/12gZcH1Jr1xWWRzyH3yMqJ9DV_3r1HsnK/view?usp=sharing

The assignment has two files-

1. `main.js` - Simple Node/Express REST API that returns a "Hello, World!" response on port 3000
2. `package.json` - Dependency management file

Step 2: Check if Node.js is installed on your computer, if not then install it.

Step 3: Run `npm install`, it will generate `node_modules` and `package-lock.json`

Step 4: Run `node main.js`, the API can now be accessed on <http://localhost:3000> and you'll see the response "Hello, World!".

Step 5: Create a `Dockerfile` to build an image of the above code.

Step 6: Run a container using the created image. (Don't forget to expose the required port(s)).

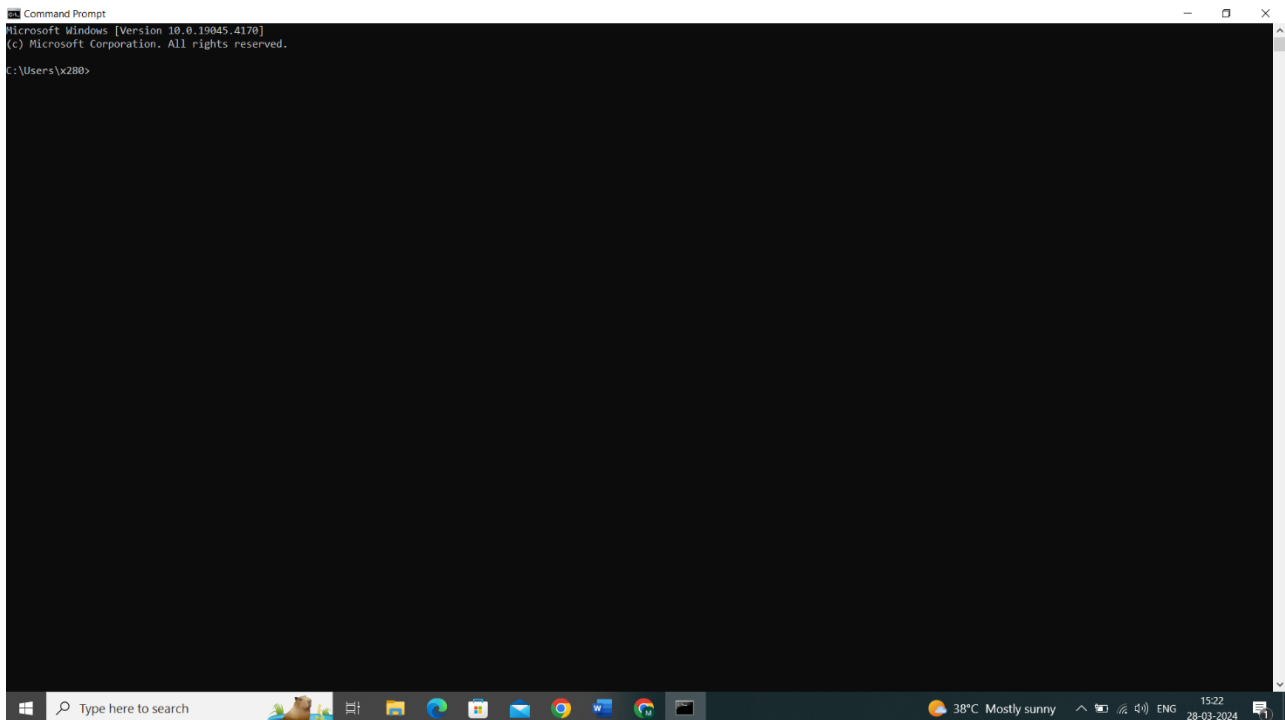
Step 7: Access the API and you must see the response "Hello World!".

BONUS: Write a `docker-compose.yml` file and use `docker-compose` to complete the assignment.

Once the assignment has been completed, push it on GitHub and share the repo link with us.

NOTE: You can install Docker on your system and do the assignment, or you can use an online Docker Playground on <https://labs.play-with-docker.com>.

- Download main.js and package.js files from the above link.
- Now install node.js in your computer link for download-<https://nodejs.org/en/download>
- open nodejs command prompt



command prompt looks like above

- Now change directory and go to hello-node directory.
- Cd Desktop
- Cd hello-node
- Now install npm tool in nodejs command prompt.
- Command- npm install
- Now node main.js
- Listing on port 3000

```
Command Prompt - node main.js
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\x280>npm install
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path C:\Users\x280\package.json
npm ERR! errno -4058
npm ERR! enoent Could not read package.json: Error: ENOENT: no such file or directory, open 'C:\Users\x280\package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in: C:\Users\x280\AppData\Local\npm-cache\_logs\2024-03-28T09_56_03_898Z-debug-0.log

C:\Users\x280>cd Desktop

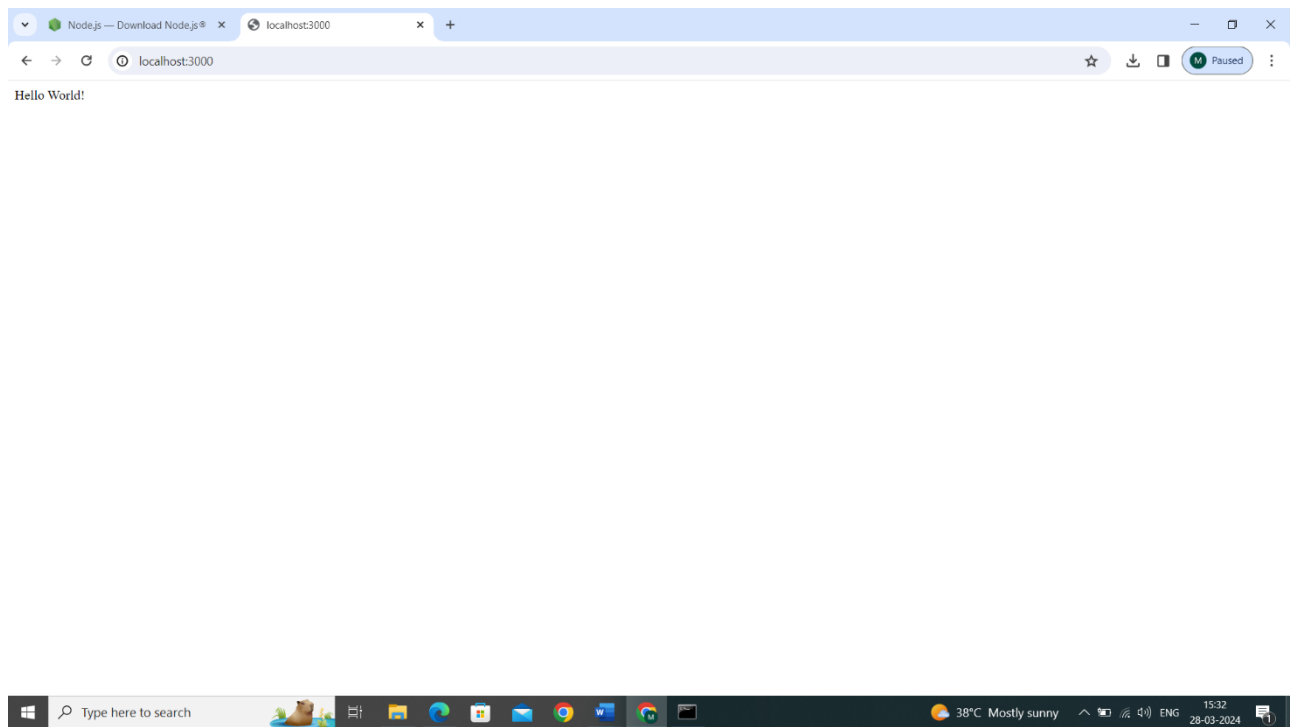
C:\Users\x280\Desktop>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\x280\Desktop>cd hello-node

C:\Users\x280\Desktop\hello-node>npm install
up to date, audited 65 packages in 1s
12 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

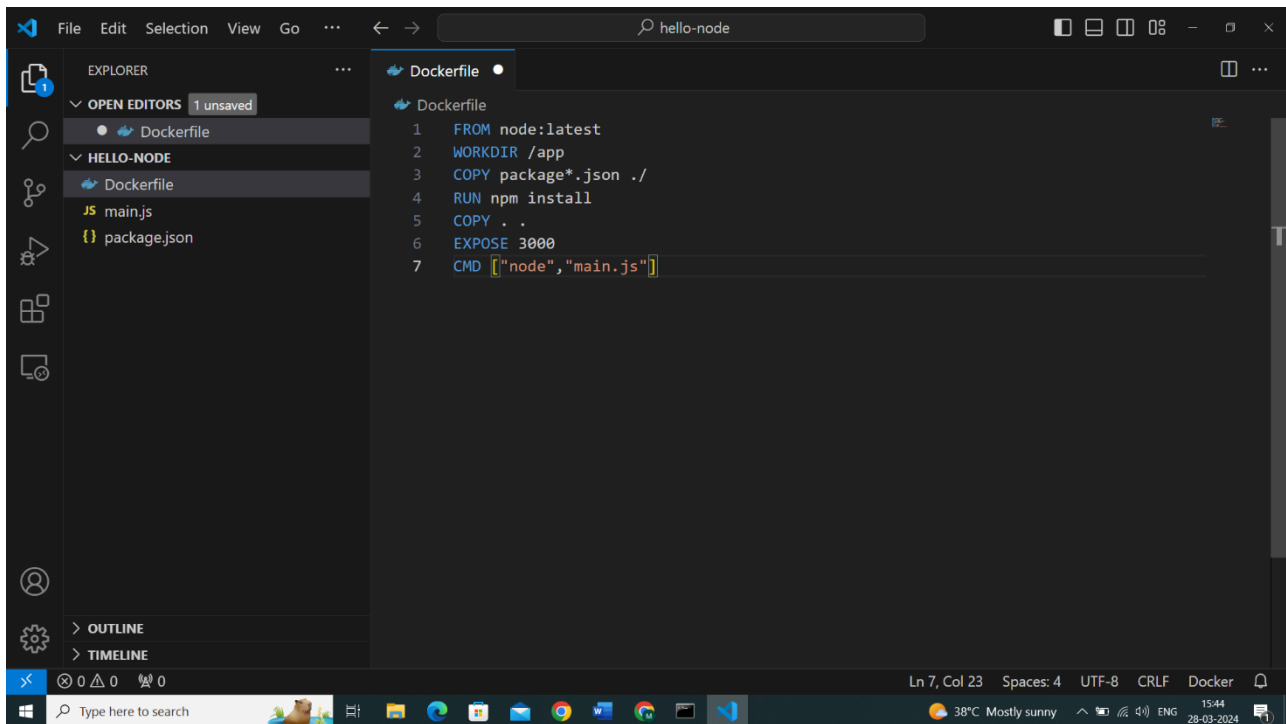
C:\Users\x280\Desktop\hello-node>node main.js
Example app listening on port 3000
```

local host on port 3000 listing



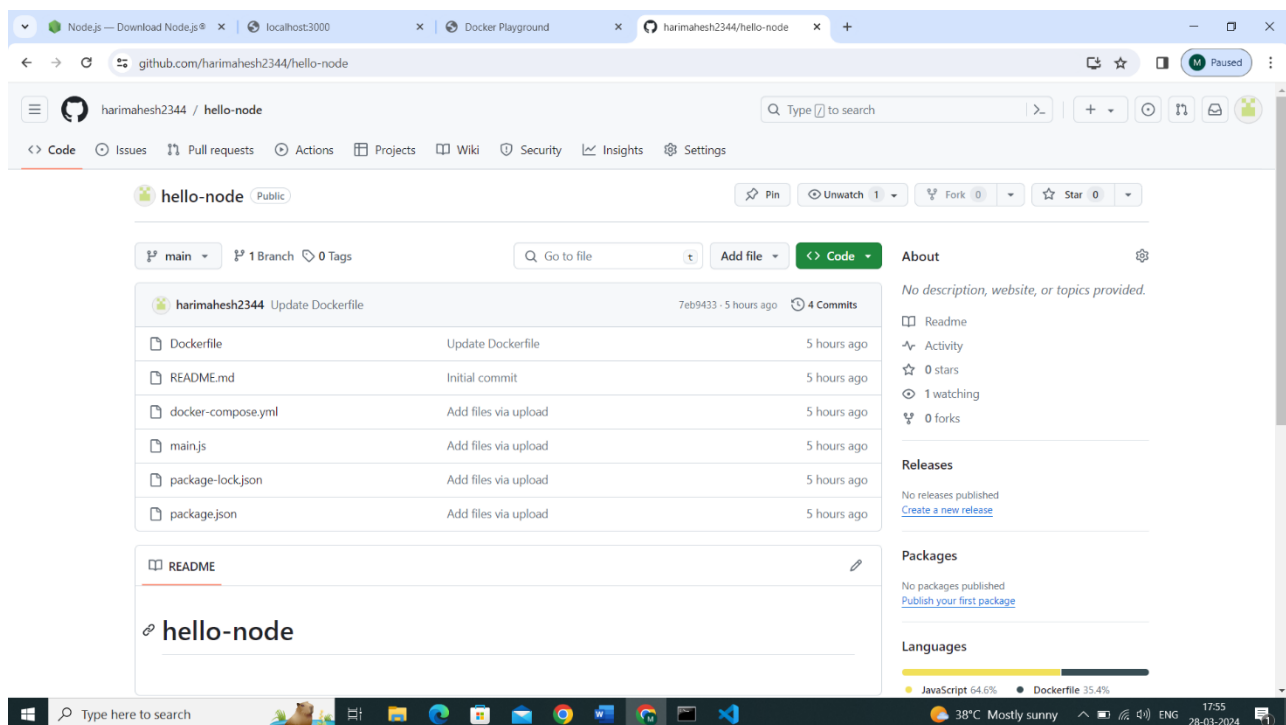
response is Hello World is successful

- Now writing Dockerfile
- Using visual studio code

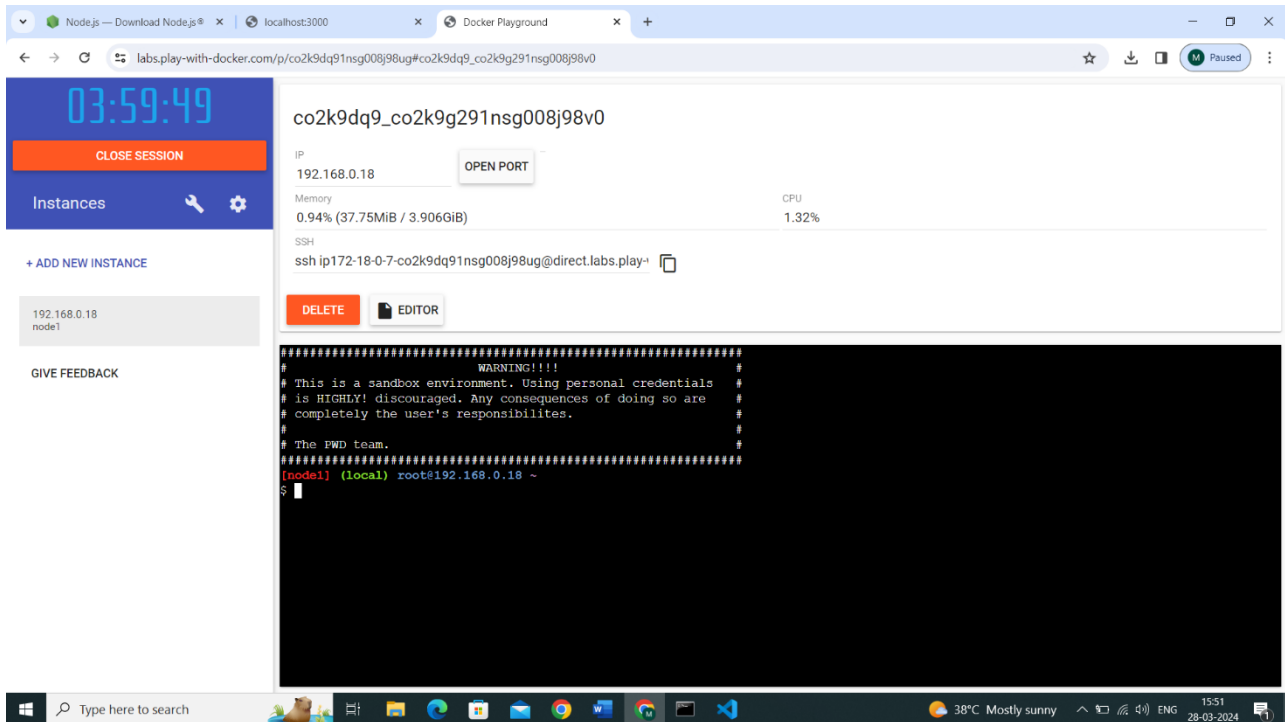


Writing Dockerfile for given files

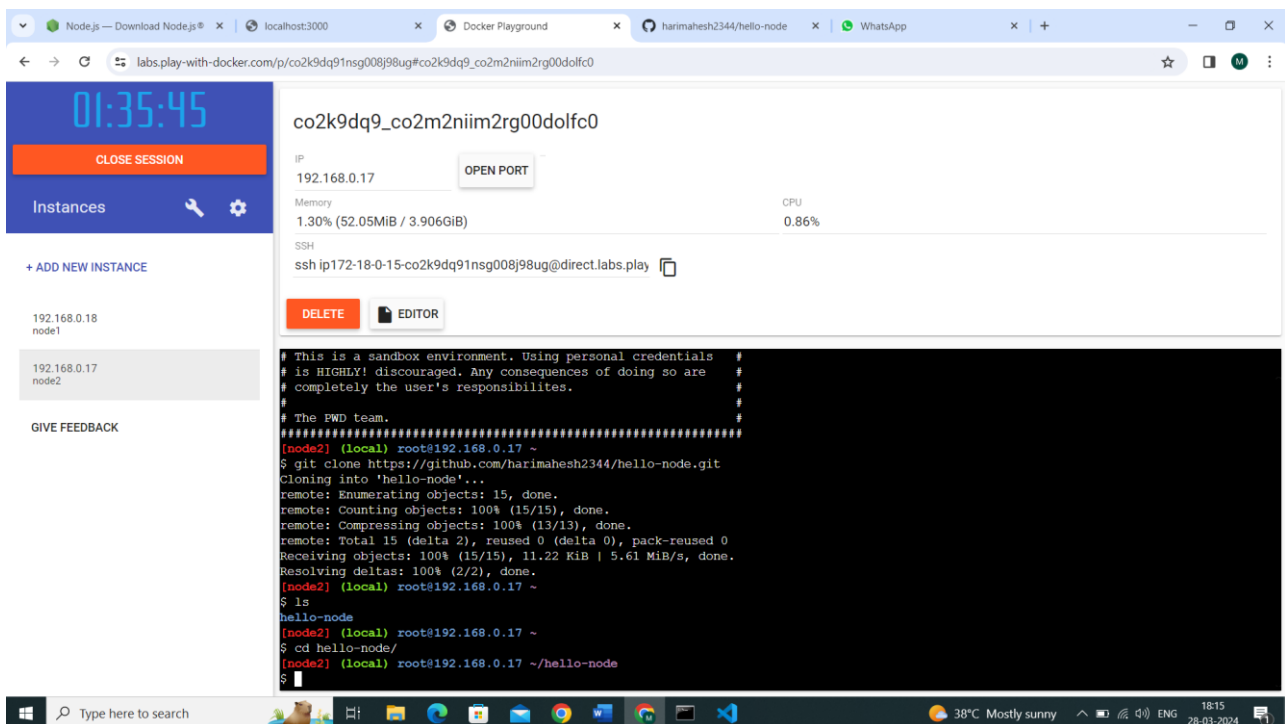
- Now add files to GitHub



- Now open docker playground



- Git clone to download files from remote repository.



Cloning from GitHub

- Cd hello-node
- Now build image using command- docker build -t <image name> .

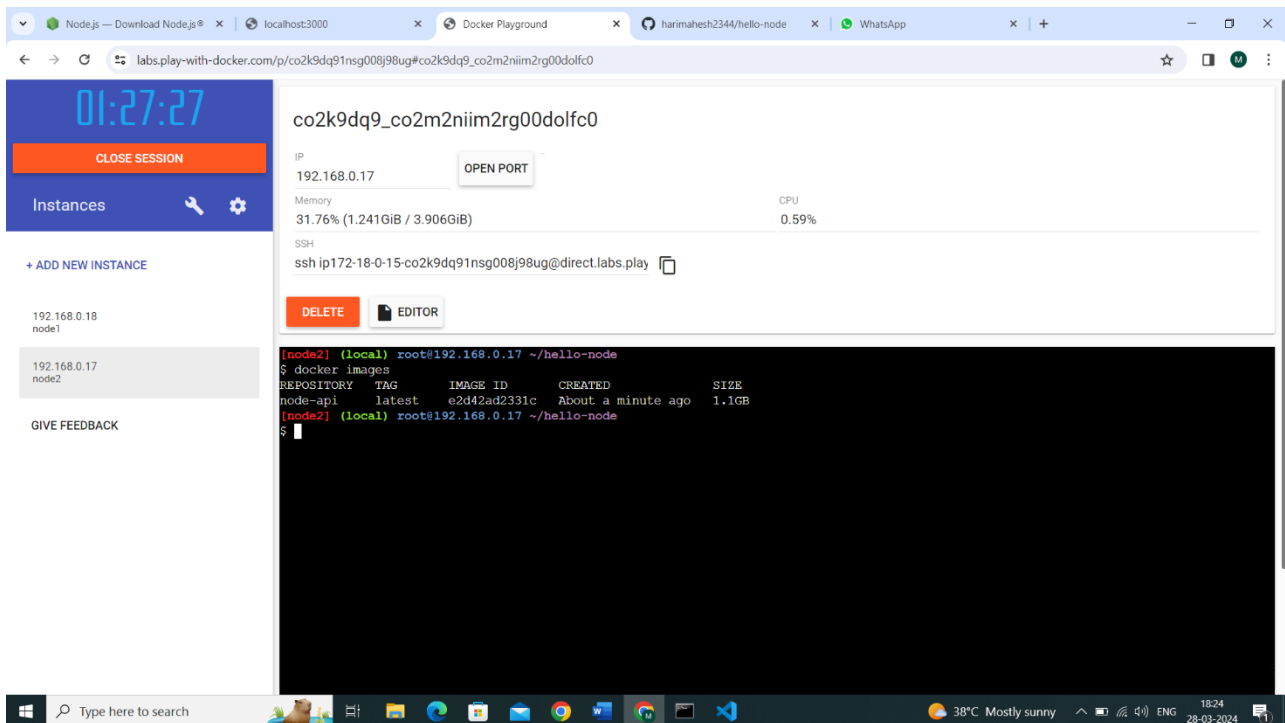
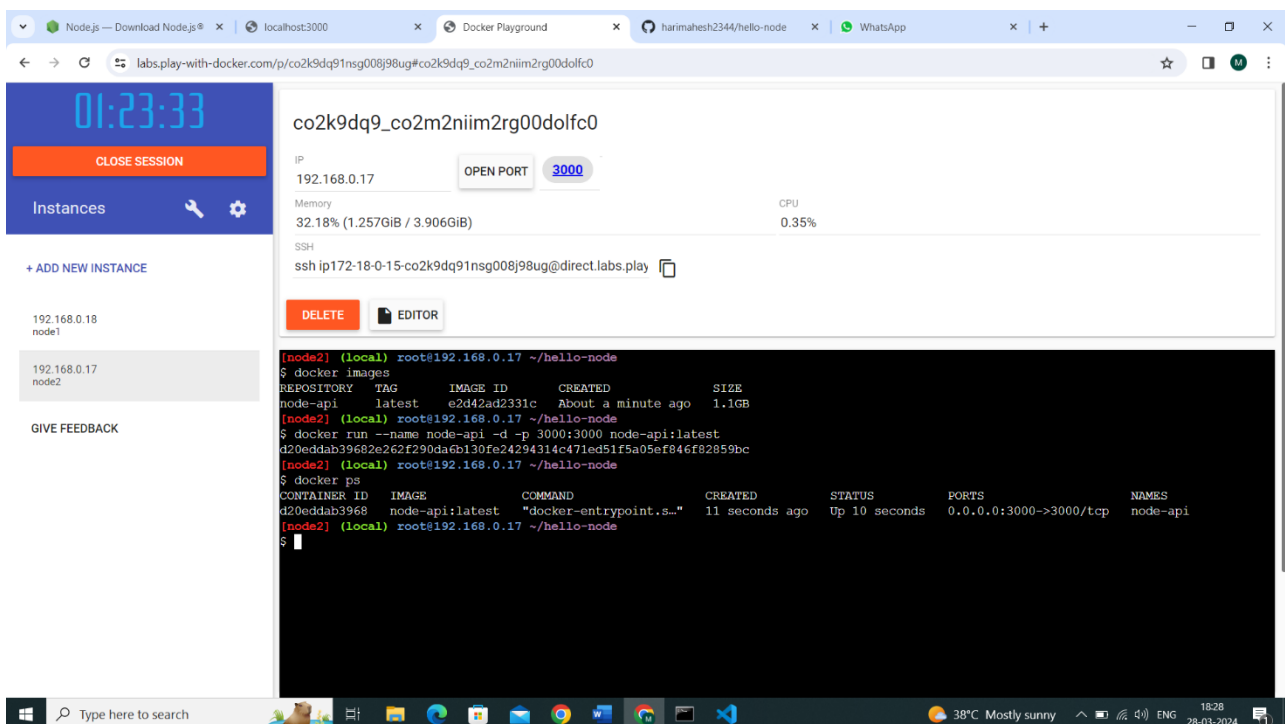
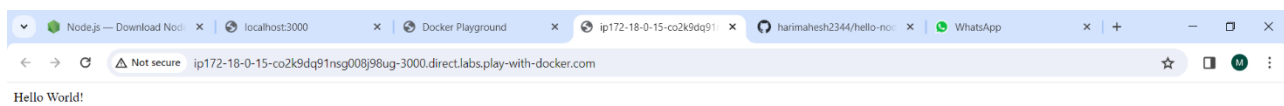


Image created successfully

- Run as container command- `docker run --name <container name> -d -p 3000:3000 <image name>:<tag>`

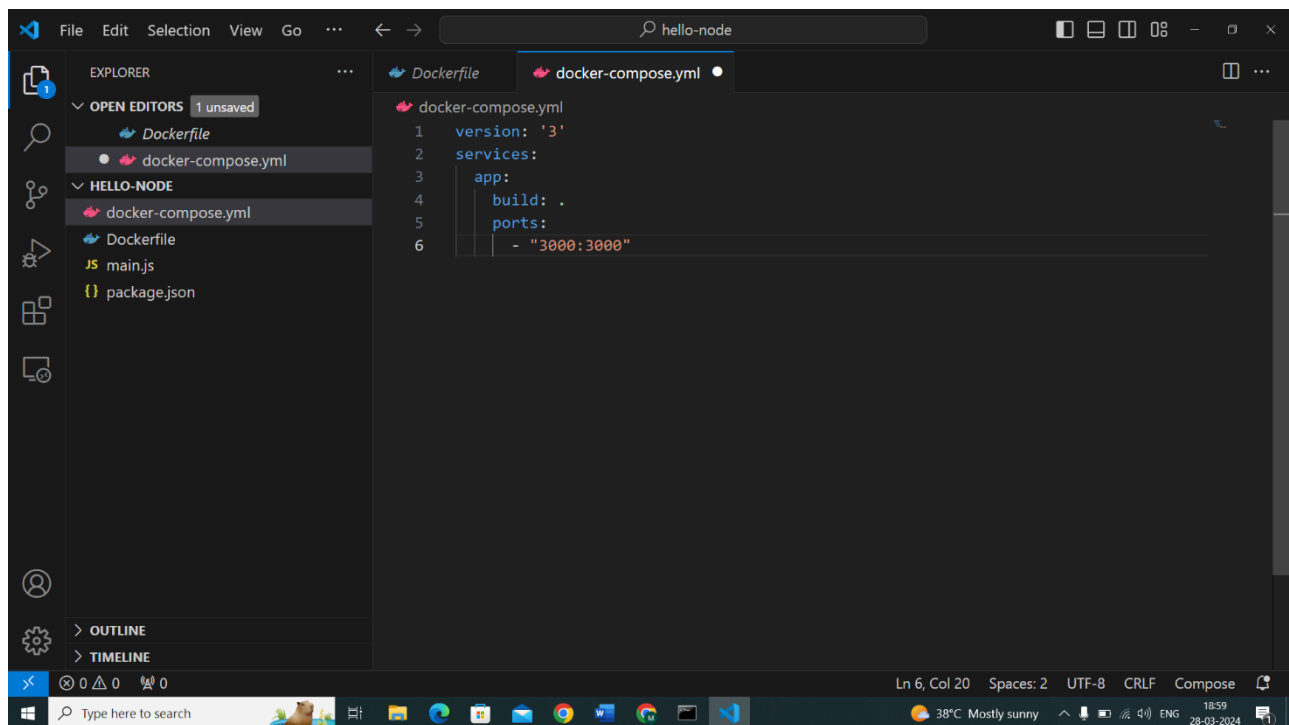


Container run successfully



with docker playground successfully hosted

- Now write docker-compose.yml using visual studio code.



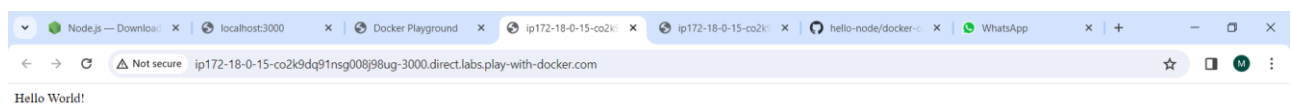
Writing docker-compose.yml in visual studio code

- Again clone into docker playground.
- Now command- docker compose up

```

(node2) (local) root@192.168.0.17 ~/hello-node
$ docker run --name node-api -d -p 3000:3000 node-api:latest
d20eddab39682e262f290da6b130fe24294314c471ed51f5a05ef846f82859bc
(node2) (local) root@192.168.0.17 ~/hello-node
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
d20eddab3968   node-api:latest "docker-entrypoint.s..." 11 seconds ago Up 10 seconds 0.0.0.0:3000->3000/tcp   node-api
(node2) (local) root@192.168.0.17 ~/hello-node
$ docker compose up
[+] Building 0.3s (10/10) FINISHED                                docker:default
=> [app internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 157B                               0.0s
=> [app internal] load .dockerignore                             0.0s
=> => transferring context: 2B                                       0.0s
=> [app internal] load metadata for docker.io/library/node:latest 0.2s
=> [app 1/5] FROM docker.io/library/node:latest@sha256:b9ccc4aca32eebf124e0ca0fd573dacffba2b9236987a1d4d2625ce3c162ecc8 0.0s
=> [app internal] load build context                             0.0s
=> => transferring context: 1.98kB                                    0.0s
=> CACHED [app 2/5] WORKDIR /app                                  0.0s
=> CACHED [app 3/5] COPY package*.json ./                        0.0s
=> CACHED [app 4/5] RUN npm install                               0.0s
=> CACHED [app 5/5] COPY . .                                       0.0s
=> [app] exporting to image                                       0.0s
=> => exporting layers                                              0.0s
=> => writing image sha256:9105d818a85f3a4785b804a0420b77071ba2621d0c2edbc552d04b7d7a93872d 0.0s
=> => naming to docker.io/library/hello-node-app                 0.0s
[+] Running 2/2
✔ Network hello-node_default Created                               0.1s
✔ Container hello-node-app-1 Created                               0.1s
Attaching to hello-node-app-1
Error response from daemon: driver failed programming external connectivity on endpoint hello-node-app-1 (3913dd02af4d1a407467b628fec5ee7f8c383367639dd894a
nd for 0.0.0.0:3000 failed: port is already allocated
(node2) (local) root@192.168.0.17 ~/hello-node
$
  
```

Docker Compose up successfully



- Host on 3000
- Successfully completed all the steps.

Thanking you