

Name : Nisarg .k. Amlani

Roll : Ce001

Id : 22ceueg082

Docker Assignment

=> Docker File

Use an official Node.js runtime as a parent image
FROM node:18

Set the working directory inside the container
WORKDIR /app

Copy package.json and package-lock.json first for efficient caching
COPY package*.json ./

Install dependencies
RUN npm install

Copy the rest of the application files
COPY . .

Expose the port the app runs on
EXPOSE 3000

Command to run the application
CMD ["node", "server.js"]

=> Server.js File

```
const express = require("express");
const app = express();
const port = process.env.PORT || 3000;

// Default route
app.get("/", (req, res) => {
  res.send("Welcome to the Dockerized Node.js App!");
});

// Custom route returning JSON
app.get("/api/data", (req, res) => {
  res.json({
    message: "Hello from the API!",
    status: "success",
    timestamp: new Date().toISOString(),
  });
});

// Route with dynamic parameter
app.get("/hello/:name", (req, res) => {
  const name = req.params.name;
  res.send(`Hello, ${name}!`);
});

// Route returning HTML
app.get("/html", (req, res) => {
  res.send(`
    <h1>Welcome to My Express App</h1>
    <p>This is a simple response with HTML content.</p>
  `);
});

// Start the server
app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});
```

=> Building Docker Image

```
PS D:\CE001\Lab_12> docker build -t node-docker-app-2 .
[+] Building 1.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:18
=> [internal] load build context
=> => transferring context: 42.37kB
=> [1/5] FROM docker.io/library/node:18@sha256:ba756f198b4b1e0114b53b23121c8ae27f7ae4d5d95ca4a0554b0649cc9c7dcf
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY package*.json ./
=> CACHED [4/5] RUN npm install
=> [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:52476640b3c939402eff797b9841e757718f8cd8071dd7f69f4e40c7b5cf2dd8
=> => naming to docker.io/library/node-docker-app-2
PS D:\CE001\Lab_12>
```

CyberCoder Improve Code Share Code Link In 37, Col 1 Spaces: 4 UTF-8

=> Docker Images

```
PS D:\CE001\Lab_12> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
node-docker-app-2	latest	52476640b3c9	About a minute ago	1.09GB
node-docker-app	latest	77b5525f5bdd	6 minutes ago	1.09GB
counter-image	latest	d2220e5d88f1	46 hours ago	218MB
ubuntu	latest	a04dc4851cbc	6 weeks ago	78.1MB
ubuntu	jammy	a24be041d957	6 weeks ago	77.9MB
mysql	latest	5568fddd4f66	7 weeks ago	797MB
kunj199/kunjit	latest	a9da46d2359b	19 months ago	1.01GB
kunj199/kunjit3	latest	406df6083fc0	19 months ago	153MB
kunj199/kunjit2	latest	c307bfa24b67	19 months ago	153MB
kunj199/kunjit1	latest	33fd4664ca26	19 months ago	153MB
<none>	<none>	b5d82b5e1e91	19 months ago	153MB

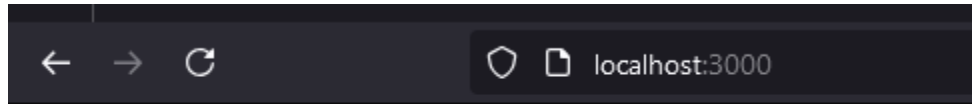
```
PS D:\CE001\Lab_12>
```

=> Running the image

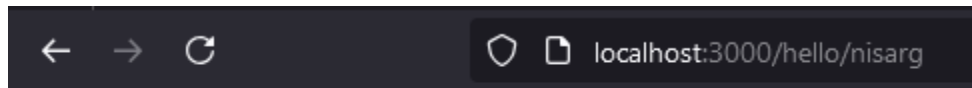
```
PS D:\CE001\Lab_12> docker run -p 3000:3000 node-docker-app-2
Server running on port 3000

```

=> Output



Welcome to the Dockerized Node.js App!

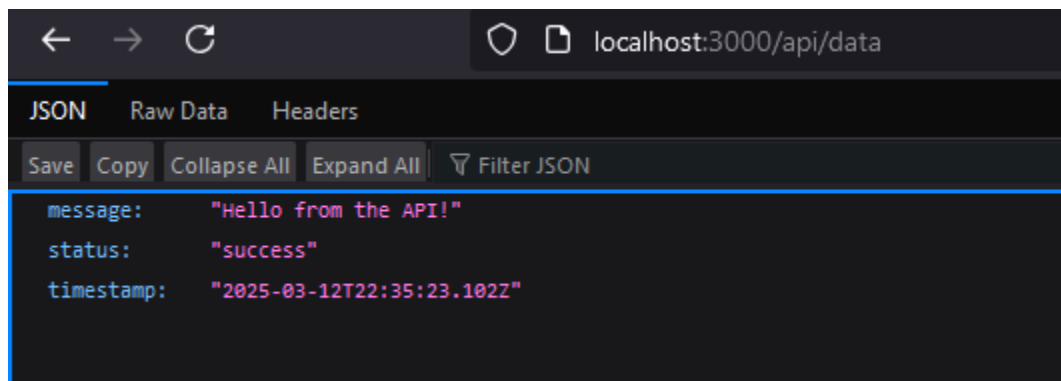


Hello, nisarg!



Welcome to My Express App

This is a simple response with HTML content.



=> Building the Image to upload in docker Hub

```
PS D:\CE001\Lab_12> docker build -t nisarg155/node-express-image .
[+] Building 2.3s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 32B                                                0.0s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/node:18                      1.9s
=> [auth] library/node:pull token for registry-1.docker.io                     0.0s
=> [internal] load build context                                                 0.1s
=> => transferring context: 41.51kB                                              0.0s
=> [1/5] FROM docker.io/library/node:18@sha256:ba756f198b4b1e0114b53b23121c8ae27f7ae4d5d95ca4a0554b0649cc9c7dcf 0.0s
=> CACHED [2/5] WORKDIR /app                                                    0.0s
=> CACHED [3/5] COPY package*.json ./                                           0.0s
=> CACHED [4/5] RUN npm install                                                  0.0s
=> CACHED [5/5] COPY . .                                                        0.0s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.0s
=> => writing image sha256:52476640b3c939402eff797b9841e757718f8cd8071dd7f69f4e40c7b5cf2dd8 0.0s
=> => naming to docker.io/nisarg155/node-express-image                         0.0s
```

=> Pushing the image to docker hub

```
PS D:\CE001\Lab_12> docker push nisarg155/node-express-image
Using default tag: latest
The push refers to repository [docker.io/nisarg155/node-express-image]
17a9a92166d4: Pushed
8101b89fac69: Pushed
33876b1b1242: Pushed
d25684c4bbd5: Pushed
f270deee6385: Mounted from library/node
e129ba4f1574: Mounted from library/node
780658909072: Mounted from library/node
eae0a84b6ca2: Mounted from library/node
4b017a36fd9c: Mounted from library/node
20a9b386e10e: Mounted from library/node
f8217d7865d2: Mounted from library/node
01c9a2a5f237: Mounted from library/node
latest: digest: sha256:f61cc6712380b0c610a4bd0539017c3246e24f09b9bc1d7f3fcf4a2ce85c7d5b size: 2840
PS D:\CE001\Lab_12>
```