

# Dockerization of NodeJS Application

**Author:** Oluwaseun Osunsola

**Environment & Tools:** Windows, Docker Desktop, VSCode

**Project Link:** <https://github.com/Oluwaseunoa/DevOps-Projects/tree/main/Docker-Projects/Dockerization%20of%20NodeJS%20Application%20-%20v0.1>

## ❖ Project Overview

This project demonstrates how to **containerize a simple Node.js (Express) application using Docker**. The goal is to show a standard workflow for building, running, and accessing a Node.js application inside a Docker container.

The application exposes a single HTTP endpoint (/) that returns a welcome message.

## 📁 Project Structure

```
Dockerization of NodeJS Application - v0.1
  Dockerfile
  package.json
  README.md

  src
    server.js
```

## ⚙️ Application Details

### server.js

- Built with **Express.js**
- Listens on **port 3000**
- Responds with a welcome message on the root endpoint (/)

```
const express = require('express');
const app = express();

app.get('/', (req,res)=>{
    res.send("Welcome to my awesome app!");
});

app.listen(3000, function () {
    console.log("app listening on port 3000");
});
```

## Docker Configuration

### Dockerfile Explanation

```
FROM node:19-alpine  
  
COPY package.json /app/  
COPY src /app/  
  
WORKDIR /app  
RUN npm install  
  
CMD ["node", "server.js"]
```

#### Breakdown:

- Uses a lightweight **Node 19 Alpine** base image
  - Copies application files into the container
  - Installs dependencies using `npm install`
  - Starts the application using Node.js
- 

## Build & Run Instructions

### 1 Build the Docker Image

Run the following command from the project root directory:

```
docker build -t node-app:1.0 .
```

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files: Dockerfile, server.js, README.md.
- Terminal:** Displays the command `docker build -t node-app:1.0 .` followed by the build logs. The logs show the process of extracting files from Docker images and copying them into the build context. The total build time is approximately 2.9 seconds.
- Status Bar:** Shows the current file is Dockerfile, and other status indicators like line 5, column 1, and file size.

```

FROM node:19-alpine
COPY package.json /app/
COPY src /app/
WORKDIR /app
RUN npm install
CMD ["node", "server.js"]

package.json README.md src

HP@DESKTOP-I9M74R1 MINGW64 ~\Documents\Workspace\DevOps-Projects\Docker-Projects\Dockerization of NodeJS Application - v0.1 (main)
$ docker build -t node-app:1.0 .
=> sha256:1197750296b3abe1d21ffbb3d3ea76df5ba887cf82c8e3284d267ccb2aa1724a 48.15MB / 48.15MB 64.7s
=> sha256:8a49fdb3b6a5ff2bd8ec6a86c05b2922a0f7454579ecc07637e94dfd1d0639b6 3.40MB / 3.40MB 14.9s
=> => extracting sha256:8a49fdb3b6a5ff2bd8ec6a86c05b2922a0f7454579ecc07637e94dfd1d0639b6 0.4s
=> => extracting sha256:1197750296b3abe1d21ffbb3d3ea76df5ba887cf82c8e3284d267ccb2aa1724a 5.8s
=> => extracting sha256:f352bc07f19b43a8678cc8c8efeb162ccb6193ead7af6dd366639a01402d1819e 0.2s
=> => extracting sha256:47be83a79857fb67c4d144471b8301ae6fb874971bfaa60d12dc97ea1355cff 0.1s
=> [internal] load build context 0.2s
=> => transferring context: 447B 0.1s
=> [2/5] COPY package.json /app/ 0.5s
=> [3/5] COPY src /app/ 0.1s
=> [4/5] WORKDIR /app 0.1s
=> [5/5] RUN npm install 94.9s
=> exporting to image 2.9s

```

## Docker image build process

## ② Run the Docker Container and Confirm

```
docker run -d -p 3000:3000 node-app:1.0
docker ps
```

### Explanation:

- `-d` → Run container in detached mode
- `-p 3000:3000` → Bind container port 3000 to host port 3000
- `--name` → Assigns a readable container name
- `docker ps` → Lists all running containers

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files: Dockerfile U, server.js U, README.md U.
- Terminal:** Displays the command `docker run -d -p 3000:3000 node-app:1.0` and its output: \$ docker ps, showing a container with ID 8c096826ede0595d9e153d722ea05ff074dfde1235499583d33344ad4507c693.
- Output:** Shows the command \$ docker ps and its output, listing the container with ID 8c096826ede0.

Running container output / `docker ps`

## 🌐 Accessing the Application

Once the container is running, open your browser and navigate to:

```
http://localhost:3000
```

You should see:

```
Welcome to my awesome app!
```



Browser showing the running application on port 3000

## 📝 Verification Commands (Optional)

```
docker ps  
docker logs nodejs-app
```

The screenshot shows a code editor interface with several tabs open: Dockerfile, server.js, and README.md. The README.md file contains the following text:

```
## 🖋 Verification Commands (Optional)  
129 ## 🖋 Verification Commands (Optional)  
130  
131 ````bash  
132 docker ps  
133 docker logs nodejs-app  
134 ````  
135  
136 ---  
137  
138 ## 🔴 Stopping & Removing the Container
```

Below the code editor is a terminal window with the following history:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Docker-Projects/Dockerization of NodeJS Application - v0.1 (main)  
$ docker logs  
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Docker-Projects/Dockerization of NodeJS Application - v0.1 (main)  
$ docker logs 8c096826ede0  
app listening on port 3000
```

Checking container logs

## ⓧ Stopping & Removing the Container

```
docker stop 8c096826ede0 #Container <id>  
docker rm 8c096826ede0 #Container <id>
```

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows 'Dockerfile U' and 'server.js U'.
- Search Bar:** 'DevOps-Projects'.
- Code Editor:** README.md file containing code related to Dockerization of NodeJS Application.
- Terminal:** Shows the command history for stopping and removing a Docker container:

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Docker-Projects/Dockerization of NodeJS Application - v0.1 (main)
$ docker stop 8c096826ede0
8c096826ede0

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Docker-Projects/Dockerization of NodeJS Application - v0.1 (main)
$ docker rm 8c096826ede0
8c096826ede0

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Docker-Projects/Dockerization of NodeJS Application - v0.1 (main)
$
```

Bottom status bar: Line 142, Col 23 (12 selected), Spaces: 4, UTF-8, CRLF, {}, Markdown, Prettier.

## Stopping and removing container

## 📦 Technologies Used

- Node.js
- Express.js
- Docker

## 📝 Notes

- Ensure Docker is installed and running before building the image
- Port **3000** must be free on the host machine

## 📄 Version

**v0.1** – Initial Dockerization of Node.js application

## 👤 Author

Oluwaseun Osunsola