1. DOCKER INSTALL

sudo apt install docker.io

sudo systemctl start docker

sudo systemctl enable docker

sudo systemctl status docker

2. INSTALL NPM

Node.js comes with `npm` (Node Package Manager) by default. Install it using the following command:

sudo apt install nodejs npm

Verify the installation:
Check the installed versions of Node.js and npm:

node -v
npm -v

3. INSTALL AWS CLI AND MYSQL-CLIENT

Install AWS CLI and do aws configure

aws configure

Install mysql client

sudo apt install mysql-client -y

4. CREATE MYSQL DATABASE (RDS)

Login into database using endpoint and create database

mysql -h rdsendpoint -u admin -p

Create database

```
USE userdb;
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    password VARCHAR(100)
);
```

5. Create server.js, index.html, Dockerfile

NOTE: change Ip address(EC2) in serevr.js and index.html

server.js

```
const express = require('express');
const mysql = require('mysql2');
const bodyParser = require('body-parser');
const cors = require('cors');
const path = require('path'); // Add this line
const app = express();
const port = 3000;

// Middleware
app.use(bodyParser.json());
app.use(cors());

// Serve static files from the "public" folder
app.use(express.static(path.join(__dirname, 'public'))); // Add this line

// MySQL Database Connection
const db = mysql.createConnection({
    host: process.env.DB_HOST, // Get the RDS host from environment variable
    user: process.env.DB_USER, // Get the DB username from environment variable
    password: process.env.DB_PASSWORD, // Get the DB password from environment variable
    database: process.env.DB_NAME // Get the DB name from environment variable
});

// Connect to MySQL
db.connect(err => {
    if (err) {
        console.error('Error connecting to the database:', err.stack);
        return;
    }
    console.log('Connected to MySQL database');
});

// Register User Route (POST request)
app.post('/register', (req, res) => {
```

```
    const { name, email, password } = req.body;

    if (!name || !email || !password) {
        return res.status(400).json({ message: 'All fields are required' });
    }

    const query = 'INSERT INTO users (name, email, password) VALUES (?, ?, ?)';
    db.query(query, [name, email, password], (err, result) => {
        if (err) {
            return res.status(500).json({ message: 'Error registering user' });
        }
        res.status(201).json({ message: 'User registered successfully', id: result.insertId });
    });
});

// Start the server
app.listen(port, () => {
    console.log(`Server is running on http://13.233.105.126:${port}`);
});
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Registration</title>
</head>
<body style="font-family: 'Arial', sans-serif; background-color: #f4f4f9; display: flex; justify-
content: center; align-items: center; height: 100vh; margin: 0;">
    <div style="background-color: #ffffff; padding: 2rem; border-radius: 10px; box-shadow: 0 4px
8px rgba(0, 0, 0, 0.1); width: 300px;">
        <h1 style="color: #333; text-align: center; margin-bottom: 1.5rem;">User Registration</h1>

        <!-- Registration Form -->
        <form id="registration-form">
            <label for="name" style="display: block; margin-bottom: 0.5rem; color:
#555;">Name:</label>
            <input type="text" id="name" name="name" required style="width: 100%; padding:
0.5rem; margin-bottom: 1rem; border: 1px solid #ccc; border-radius: 5px; font-size:
1rem;"><br><br>

            <label for="email" style="display: block; margin-bottom: 0.5rem; color:
#555;">Email:</label>
            <input type="email" id="email" name="email" required style="width: 100%; padding:
0.5rem; margin-bottom: 1rem; border: 1px solid #ccc; border-radius: 5px; font-size:
1rem;"><br><br>
```

```
        <label for="password" style="display: block; margin-bottom: 0.5rem; color:
#555;">Password:</label>
        <input type="password" id="password" name="password" required style="width: 100%;
padding: 0.5rem; margin-bottom: 1.5rem; border: 1px solid #ccc; border-radius: 5px; font-size:
1rem;"><br><br>

        <button type="submit" style="width: 100%; padding: 0.75rem; background-color: #007bff;
color: white; border: none; border-radius: 5px; font-size: 1rem; cursor: pointer;">
            Register
        </button>
    </form>
</div>

<script>
    // JavaScript to handle form submission
    document.getElementById('registration-form').addEventListener('submit', async (e) => {
        e.preventDefault();  // Prevents default form submission

        // Get form data
        const formData = new FormData(e.target);
        const data = {
            name: formData.get('name'),
            email: formData.get('email'),
            password: formData.get('password')
        };

        // Send POST request to Node.js backend
        const response = await fetch('http://localhost:3000/register', {
            method: 'POST',  // Ensure it's a POST request
            headers: {
                'Content-Type': 'application/json'  // Specify content type as JSON
            },
            body: JSON.stringify(data)  // Send the form data as JSON
        });

        // Handle the response from the backend
        const result = await response.json();
        if (response.ok) {
            alert('Registration Successful!');
        } else {
            alert('Registration Failed: ' + result.message);
        }
    });
</script>
</body>
</html>
```

Dockerfile

```
# Use the official Node.js image
FROM node:16

# Set the working directory
WORKDIR /usr/src/app

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

# Install dependencies
RUN npm install

# Copy the rest of the application
COPY . .

# Expose the application port
EXPOSE 3000

# Start the application
CMD ["node", "server.js"]
```

6. INSTALL DEPENDENCY

Initialize a Node.js project:
Run the following command to generate a `package.json` file:

```
npm init -y
```

This will create a `package.json` file with default values.

Install required dependencies:
Install the dependencies needed for your Node.js application:

```
npm install express mysql2 body-parser cors dotenv
```

This will:

- Install the packages.
- Generate a `package-lock.json(which locks the dependency versions).`

7. BUILD DOCKERFILE AND CREATE CONTAINER

Run the Application

1. Build the Node.js Docker image:

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

2. Run the Node.js container and link it to the MySQL container:

   ```
   docker run \
   --name node-app \
   -e DB_HOST=database-1.cbu4g6i02din.ap-south-1.rds.amazonaws.com \
   -e DB_USER=admin \
   -e DB_PASSWORD=admin123 \
   -e DB_NAME=userdb \
   -p 3000:3000 \
   -d node-app
   ```

3. Access the application:
   - Open your browser and go to `http://your-ec2-public-ip:3000`.
   - Fill out the registration form and submit it.