create a nodejs simple appn and then we push code on github after this start deploying appn on aws ec2 - step by step from scratch please - let's start - please step by step from scratch - just a simple nodejs simple appn

**ChatGPT said:**

Let's start step by step.

**Step 1: Set Up a Simple Node.js Application**

**1. Install Node.js**

Ensure you have Node.js installed. You can check by running:

bash

CopyEdit

node -v

npm -v

If not installed, download and install it from [Node.js official site](https://nodejs.org/).

**2. Create a Project Directory**

bash

CopyEdit

mkdir simple-node-app && cd simple-node-app

**3. Initialize a Node.js Project**

bash

CopyEdit

npm init -y

This will create a package.json file.

**4. Install Express.js (for a simple web server)**

bash

CopyEdit

npm install express

**5. Create index.js and Add the Code**

Create a file named index.js in the project directory and add the following code:

javascript

CopyEdit

const express = require('express');

const app = express();

const PORT = process.env.PORT || 3000;

app.get('/', (req, res) => {

res.send('Hello, World! This is a simple Node.js app.');

});

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});

**6. Run the Application Locally**

bash

CopyEdit

node index.js

Visit http://localhost:3000/ in your browser. You should see:  
**"Hello, World! This is a simple Node.js app."**

**Next Step: Push Code to GitHub**

Shall we continue? 🚀

4o

**You said:**

continue please

**ChatGPT said:**

**Step 2: Push Code to GitHub**

**1. Initialize a Git Repository**

Run the following command inside your project directory:

bash

CopyEdit

git init

**2. Create a .gitignore File**

Create a .gitignore file and add the following lines to exclude unnecessary files:

bash

CopyEdit

node\_modules

.env

Run:

bash

CopyEdit

echo "node\_modules\n.env" > .gitignore

**3. Add and Commit Files**

bash

CopyEdit

git add .

git commit -m "Initial commit: Simple Node.js App"

**4. Create a GitHub Repository**

* Go to [GitHub](https://github.com/)
* Click on New Repository
* Name it **simple-node-app**
* Do NOT initialize with README (we already have files)
* Copy the repository URL

**5. Link Local Repo to GitHub**

Replace <your-github-repo-url> with your actual repo URL:

bash

CopyEdit

git remote add origin <your-github-repo-url>

**6. Push Code to GitHub**

bash

CopyEdit

git branch -M main

git push -u origin main

Your code is now on GitHub. 🎉

**Next Step: Deploy on AWS EC2**

Shall we continue? 🚀

**You said:**

continue please

**ChatGPT said:**

**Step 3: Deploy Node.js App on AWS EC2**

Now, we will set up an EC2 instance and deploy our Node.js application.

**1. Launch an EC2 Instance**

**1.1 Login to AWS Console**

* Go to [AWS EC2 Console](https://console.aws.amazon.com/ec2/)
* Click **Launch Instance**
* Choose **Amazon Linux 2** (or Ubuntu 22.04)
* Choose **t2.micro** (Free Tier Eligible)
* Click **Next → Next → Configure Security Group**
* Add a rule:
  + **Type:** HTTP
  + **Port:** 80
  + **Source:** Anywhere (0.0.0.0/0)
* Click **Launch**
* **Download & Save the Key Pair (.pem file)**
  + Example name: mykey.pem

**2. Connect to EC2 Instance**

**2.1 Move Key to a Secure Location**

Move the .pem file to a secure location and set permissions:

bash

CopyEdit

mv ~/Downloads/mykey.pem ~/.ssh/

chmod 400 ~/.ssh/mykey.pem

**2.2 SSH into EC2**

Replace your-ec2-public-ip with your instance's Public IP:

bash

CopyEdit

ssh -i ~/.ssh/mykey.pem ec2-user@your-ec2-public-ip

For Ubuntu instances, use:

bash

CopyEdit

ssh -i ~/.ssh/mykey.pem ubuntu@your-ec2-public-ip

**3. Install Node.js on EC2**

For **Amazon Linux 2**, run:

bash

CopyEdit

sudo yum update -y

sudo yum install -y gcc-c++ make

curl -fsSL https://rpm.nodesource.com/setup\_18.x | sudo bash -

sudo yum install -y nodejs

For **Ubuntu**, run:

bash

CopyEdit

sudo apt update -y

sudo apt install -y nodejs npm

Verify installation:

bash

CopyEdit

node -v

npm -v

**4. Clone the GitHub Repository**

bash

CopyEdit

git clone https://github.com/<your-github-username>/simple-node-app.git

cd simple-node-app

**5. Install Dependencies and Start the App**

bash

CopyEdit

npm install

node index.js

Your app is now running on port 3000.

**6. Keep the App Running with PM2**

Install **PM2** to run the app in the background:

bash

CopyEdit

npm install -g pm2

pm2 start index.js --name "simple-node-app"

pm2 startup

pm2 save

**7. Set Up Firewall & Reverse Proxy with Nginx**

**7.1 Install Nginx**

For **Amazon Linux 2**:

bash

CopyEdit

sudo amazon-linux-extras enable nginx1

sudo yum install -y nginx

For **Ubuntu**:

bash

CopyEdit

sudo apt install -y nginx

**7.2 Configure Nginx**

Edit the Nginx configuration:

bash

CopyEdit

sudo nano /etc/nginx/nginx.conf

Replace:

nginx

CopyEdit

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

Save (CTRL + X, Y, Enter), then restart Nginx:

bash

CopyEdit

sudo systemctl restart nginx

sudo systemctl enable nginx

**8. Access the App in Browser**

Now, open a browser and visit:

arduino

CopyEdit

http://your-ec2-public-ip

You should see:  
**"Hello, World! This is a simple Node.js app."** 🎉

**Next Steps**

✅ **We have successfully deployed a Node.js app on AWS EC2!**

Would you like to set up a custom domain, enable HTTPS with SSL, or create a CI/CD pipeline? 🚀

4o

**You said:**

https://github.com/deepanshu1334/node-js-app.git - this is my repo

**ChatGPT said:**

Great! Now, let's deploy your Node.js app from your GitHub repository (https://github.com/deepanshu1334/node-js-app.git) to your AWS EC2 instance.

**Step 1: Connect to Your EC2 Instance**

If you're not already connected, use SSH to log in:

bash

CopyEdit

ssh -i ~/.ssh/mykey.pem ec2-user@your-ec2-public-ip

*(For Ubuntu, use ubuntu@your-ec2-public-ip instead of ec2-user.)*

**Step 2: Install Node.js and Git**

For **Amazon Linux 2**:

bash

CopyEdit

sudo yum update -y

sudo yum install -y gcc-c++ make

curl -fsSL https://rpm.nodesource.com/setup\_18.x | sudo bash -

sudo yum install -y nodejs git

For **Ubuntu**:

bash

CopyEdit

sudo apt update -y

sudo apt install -y nodejs npm git

Verify installation:

bash

CopyEdit

node -v

npm -v

git --version

**Step 3: Clone Your GitHub Repository**

Navigate to your home directory:

bash

CopyEdit

cd ~

Clone your repository:

bash

CopyEdit

git clone https://github.com/deepanshu1334/node-js-app.git

cd node-js-app

**Step 4: Install Dependencies & Start the App**

bash

CopyEdit

npm install

node index.js

Your app is now running on port **3000**.

To test, run:

bash

CopyEdit

curl http://localhost:3000

You should see:

csharp

CopyEdit

Hello, World! This is a simple Node.js app.

**Step 5: Keep the App Running with PM2**

PM2 keeps your app running in the background. Install and start:

bash

CopyEdit

npm install -g pm2

pm2 start index.js --name "node-js-app"

pm2 startup

pm2 save

**Step 6: Set Up Nginx as a Reverse Proxy**

We need Nginx to route HTTP traffic to our Node.js app.

**6.1 Install Nginx**

For **Amazon Linux 2**:

bash

CopyEdit

sudo amazon-linux-extras enable nginx1

sudo yum install -y nginx

For **Ubuntu**:

bash

CopyEdit

sudo apt install -y nginx

**6.2 Configure Nginx**

Edit the Nginx config file:

bash

CopyEdit

sudo nano /etc/nginx/nginx.conf

Replace the server {} block with:

nginx

CopyEdit

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

Save and exit (CTRL + X, then Y, then Enter).

Restart Nginx:

bash

CopyEdit

sudo systemctl restart nginx

sudo systemctl enable nginx

**Step 7: Open Firewall (if needed)**

If your app isn’t accessible, allow HTTP traffic:

bash

CopyEdit

sudo ufw allow 80

sudo ufw allow 443

sudo ufw enable

For AWS security groups:

* Go to **EC2 Dashboard** → **Security Groups**
* Edit inbound rules:
  + **Allow HTTP (80)**
  + **Allow HTTPS (443)**

**Step 8: Access Your App in Browser**

Now, visit:

arduino

CopyEdit

http://your-ec2-public-ip

You should see your app running! 🎉

**Next Steps**

✅ **Your Node.js app is live on AWS EC2!**  
Would you like to set up a **custom domain, HTTPS (SSL with Let's Encrypt), or CI/CD with GitHub Actions?** 🚀

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ npm install node index.js added 69 packages, and audited 70 packages in 2s 14 packages are looking for funding run npm fund for details found 0 vulnerabilities Server is running on http://localhost:3000 ^C ubuntu@ip-172-31-13-138:~/node-js-app$ curl http://localhost:3000 curl: (7) Failed to connect to localhost port 3000 after 0 ms: Couldn't connect to server

**ChatGPT said:**

It looks like your Node.js app started successfully but isn't accessible. Let's troubleshoot and fix this.

**Step 1: Check If the App is Running**

Since you pressed **Ctrl + C**, your app stopped. Run it again in the background:

bash

CopyEdit

node index.js &

Then, check if it's running:

bash

CopyEdit

ps aux | grep node

If it's not running, start it using:

bash

CopyEdit

pm2 start index.js --name "node-js-app"

pm2 save

pm2 status

**Step 2: Ensure the App Listens on All IPs**

Your app may be listening only on localhost. Open index.js:

bash

CopyEdit

nano index.js

Find this line:

js

CopyEdit

app.listen(3000, () => console.log(`Server is running on http://localhost:3000`));

Replace with:

js

CopyEdit

app.listen(3000, '0.0.0.0', () => console.log(`Server is running on http://0.0.0.0:3000`));

Save and exit (CTRL + X, Y, Enter), then restart:

bash

CopyEdit

pm2 restart node-js-app

**Step 3: Check If Port 3000 Is Open**

Run:

bash

CopyEdit

netstat -tulnp | grep 3000

If it's not listed, the app isn’t listening correctly. Restart it:

bash

CopyEdit

pm2 restart node-js-app

**Step 4: Allow Traffic in Firewall**

bash

CopyEdit

sudo ufw allow 3000

sudo ufw enable

sudo ufw status

**Step 5: Try Curl Again**

bash

CopyEdit

curl http://localhost:3000

If this works, your app is running!

**Step 6: Use Nginx for External Access**

We’ll set up **Nginx** to route requests to your app.

1. Install Nginx:

bash

CopyEdit

sudo apt install -y nginx

1. Edit Nginx configuration:

bash

CopyEdit

sudo nano /etc/nginx/sites-available/default

1. Replace contents with:

nginx

CopyEdit

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

1. Restart Nginx:

bash

CopyEdit

sudo systemctl restart nginx

sudo systemctl enable nginx

**Step 7: Access the App from Your Browser**

Visit:

arduino

CopyEdit

http://your-ec2-public-ip

Your app should now be accessible. Let me know if you face any issues! 🚀

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ Server is running on http://localhost:3000 ps aux | grep node ps aux | grep node ubuntu 8823 0.0 5.8 617548 56956 pts/0 Sl 16:55 0:00 node index.js ubuntu 8840 0.0 0.2 7076 2048 pts/0 S+ 17:01 0:00 grep --color=auto node ubuntu@ip-172-31-13-138:~/node-js-app$

**ChatGPT said:**

Your Node.js app is running, but it's still not accessible. Let's go step by step to fix this.

**Step 1: Ensure Your App is Listening on All IPs**

Edit index.js:

bash

CopyEdit

nano index.js

Find this line:

js

CopyEdit

app.listen(3000, () => console.log(`Server is running on http://localhost:3000`));

Replace with:

js

CopyEdit

app.listen(3000, '0.0.0.0', () => console.log(`Server is running on http://0.0.0.0:3000`));

Save and exit (CTRL + X, Y, Enter).

Restart your app:

bash

CopyEdit

pm2 restart node-js-app

Check if it's running:

bash

CopyEdit

netstat -tulnp | grep 3000

If you see something like 0.0.0.0:3000, your app is listening correctly.

**Step 2: Allow Port 3000 in Firewall**

bash

CopyEdit

sudo ufw allow 3000

sudo ufw enable

sudo ufw status

Try again:

bash

CopyEdit

curl http://localhost:3000

If this works, continue to the next step.

**Step 3: Allow Inbound Traffic in AWS Security Groups**

1. Go to **AWS EC2 Console** → **Instances**
2. Find your instance → Click on **Security Group**
3. Edit **Inbound Rules** → Click **Add Rule**
   * Type: **Custom TCP Rule**
   * Protocol: **TCP**
   * Port Range: **3000**
   * Source: **0.0.0.0/0** (for public access)
4. Click **Save Rules**

Now, test from your **local machine**:

bash

CopyEdit

curl http://your-ec2-public-ip:3000

If you get a response, your app is publicly accessible!

**Step 4: Set Up Nginx for Better Performance**

If you want to access your app via port **80**, use Nginx as a reverse proxy.

1. Install Nginx:

bash

CopyEdit

sudo apt install -y nginx

1. Configure Nginx:

bash

CopyEdit

sudo nano /etc/nginx/sites-available/default

1. Replace the contents with:

nginx

CopyEdit

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

1. Restart Nginx:

bash

CopyEdit

sudo systemctl restart nginx

sudo systemctl enable nginx

Now, open your browser and visit:

arduino

CopyEdit

http://your-ec2-public-ip

Your app should load successfully! 🎉

**Final Checks**

✅ curl http://localhost:3000 works?  
✅ curl http://your-ec2-public-ip:3000 works?  
✅ http://your-ec2-public-ip in browser works?

If not, tell me which step fails. 🚀

4o

**You said:**

const express = require('express'); const app = express(); const PORT = process.env.PORT || 3000; app.get('/', (req, res) => { res.send('Hello, World! This is a simple Node.js app by Deepanshu Kumar.'); }); app.listen(PORT, () => { console.log(Server is running on http://localhost:${PORT}); });

**ChatGPT said:**

Your code is correct, but to make it accessible from outside the EC2 instance, follow these steps carefully:

**Step 1: Ensure Your App Listens on All IPs**

Modify your index.js to explicitly listen on **0.0.0.0** instead of localhost:

js

CopyEdit

const express = require('express');

const app = express();

const PORT = process.env.PORT || 3000;

app.get('/', (req, res) => {

res.send('Hello, World! This is a simple Node.js app by Deepanshu Kumar.');

});

app.listen(PORT, '0.0.0.0', () => {

console.log(`Server is running on http://0.0.0.0:${PORT}`);

});

Save the file and restart your app.

**Step 2: Restart Your App Using PM2**

Run the following commands:

bash

CopyEdit

pm2 stop all

pm2 start index.js --name "node-js-app"

pm2 save

pm2 status

Now check if your app is running:

bash

CopyEdit

netstat -tulnp | grep 3000

If you see output like 0.0.0.0:3000, it means your app is running and listening on all IPs.

**Step 3: Allow Port 3000 in the Firewall**

Run:

bash

CopyEdit

sudo ufw allow 3000

sudo ufw enable

sudo ufw status

Then test again:

bash

CopyEdit

curl http://localhost:3000

If this works, your app is running.

**Step 4: Allow Port 3000 in AWS Security Groups**

1. **Go to AWS EC2 Console** → **Instances**
2. Select your instance → Click on **Security Group**
3. **Edit Inbound Rules** → Click **Add Rule**
   * **Type:** Custom TCP
   * **Protocol:** TCP
   * **Port Range:** 3000
   * **Source:** 0.0.0.0/0 (for public access)
4. **Save Rules**

Now, test from your local machine:

bash

CopyEdit

curl http://your-ec2-public-ip:3000

**Step 5: Setup Nginx as a Reverse Proxy (Optional, Recommended for Production)**

If you want your app to run on **port 80** (default HTTP port), configure Nginx.

1. Install Nginx:

bash

CopyEdit

sudo apt install -y nginx

1. Open Nginx configuration:

bash

CopyEdit

sudo nano /etc/nginx/sites-available/default

1. Replace the contents with:

nginx

CopyEdit

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

1. Restart Nginx:

bash

CopyEdit

sudo systemctl restart nginx

sudo systemctl enable nginx

Now, visit your EC2 instance’s public IP in a browser:

arduino

CopyEdit

http://your-ec2-public-ip

Your app should now be accessible without specifying port 3000!

**Final Checklist**

✅ curl http://localhost:3000 works inside EC2?  
✅ curl http://your-ec2-public-ip:3000 works from your local machine?  
✅ http://your-ec2-public-ip opens in the browser?

Let me know which step fails, and I’ll help you fix it! 🚀

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ node index.js node:events:495 throw er; // Unhandled 'error' event ^ Error: listen EADDRINUSE: address already in use 0.0.0.0:3000 at Server.setupListenHandle [as \_listen2] (node:net:1811:16) at listenInCluster (node:net:1859:12) at doListen (node:net:2008:7) at process.processTicksAndRejections (node:internal/process/task\_queues:83:21) Emitted 'error' event on Server instance at: at emitErrorNT (node:net:1838:8) at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) { code: 'EADDRINUSE', errno: -98, syscall: 'listen', address: '0.0.0.0', port: 3000 } Node.js v18.19.1 ubuntu@ip-172-31-13-138:~

**ChatGPT said:**

The error **"EADDRINUSE: address already in use 0.0.0.0:3000"** means that your application is already running on port **3000**, and you are trying to start another instance on the same port.

**Step 1: Find and Kill the Process Using Port 3000**

Run the following command to find the process ID (PID) using port **3000**:

bash

CopyEdit

sudo lsof -i :3000

You will see output like this:

pgsql

CopyEdit

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

node 8823 ubuntu 20u IPv4 12345 0t0 TCP \*:3000 (LISTEN)

Find the **PID** (e.g., 8823 in the example above) and kill it:

bash

CopyEdit

sudo kill -9 8823

Now, check again if the port is still in use:

bash

CopyEdit

sudo lsof -i :3000

If nothing appears, the port is now free.

**Step 2: Restart Your Node.js App Using PM2 (Recommended)**

Instead of running node index.js manually, use **PM2** to manage your app:

bash

CopyEdit

pm2 start index.js --name "node-js-app"

pm2 save

pm2 status

Now, check if your app is running:

bash

CopyEdit

curl http://localhost:3000

**Step 3: Check If Your App is Listening on the Correct IP**

Run:

bash

CopyEdit

netstat -tulnp | grep 3000

If it shows 0.0.0.0:3000, your app is running correctly.

**Step 4: Make Sure Security Group Allows Port 3000**

If your app works inside the server but not from your browser, follow these steps:

1. **Go to AWS EC2 Console**
2. **Find your EC2 instance** → **Security Groups**
3. **Edit Inbound Rules** → Add a new rule:
   * **Type:** Custom TCP
   * **Port Range:** 3000
   * **Source:** 0.0.0.0/0 (for public access)
4. **Save Rules**

Now, try accessing:

bash

CopyEdit

curl http://your-ec2-public-ip:3000

or open **http://your-ec2-public-ip:3000** in your browser.

**Step 5: (Optional) Use Nginx for Port 80 Access**

If you want to access the app without specifying port **3000**, set up an Nginx reverse proxy. Let me know if you need help with this.

**Final Checks**

✅ **Step 1:** Port 3000 is free? (sudo lsof -i :3000 returns nothing)  
✅ **Step 2:** App is running with PM2? (pm2 status)  
✅ **Step 3:** App responds on localhost? (curl http://localhost:3000)  
✅ **Step 4:** App is accessible from outside? (curl http://your-ec2-public-ip:3000)

If any step fails, let me know the error, and I'll help fix it! 🚀

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ pm2 start index.js --name "node-js-app" pm2 save pm2 status Command 'pm2' not found, did you mean: command 'tpm2' from deb tpm2-tools (5.4-1) command 'wm2' from deb wm2 (4+svn20090216-4build1) command 'pom2' from deb libpod-pom-perl (2.01-4) command 'pm' from deb powerman (2.3.27-4) command 'pmc' from deb linuxptp (4.0-1) command 'gm2' from deb gm2 (4:13.2.0-2ubuntu1) command 'pmg' from deb python3-pymatgen (2023.06.23+dfsg1-2build1) command 'pms' from deb pms (0.42-1.1) command 'pmw' from deb pmw (1:5.22-1) Try: sudo apt install <deb name> Command 'pm2' not found, did you mean: command 'gm2' from deb gm2 (4:13.2.0-2ubuntu1) command 'pmw' from deb pmw (1:5.22-1) command 'pom2' from deb libpod-pom-perl (2.01-4) command 'tpm2' from deb tpm2-tools (5.4-1) command 'pms' from deb pms (0.42-1.1) command 'pmc' from deb linuxptp (4.0-1) command 'pmg' from deb python3-pymatgen (2023.06.23+dfsg1-2build1) command 'wm2' from deb wm2 (4+svn20090216-4build1) command 'pm' from deb powerman (2.3.27-4) Try: sudo apt install <deb name> Command 'pm2' not found, did you mean: command 'wm2' from deb wm2 (4+svn20090216-4build1) command 'gm2' from deb gm2 (4:13.2.0-2ubuntu1) command 'pms' from deb pms (0.42-1.1) command 'pmw' from deb pmw (1:5.22-1) command 'tpm2' from deb tpm2-tools (5.4-1) command 'pom2' from deb libpod-pom-perl (2.01-4) command 'pmc' from deb linuxptp (4.0-1) command 'pmg' from deb python3-pymatgen (2023.06.23+dfsg1-2build1) command 'pm' from deb powerman (2.3.27-4) Try: sudo apt install <deb name> ubuntu@ip-172-31-13-138:~/node-js-app$

**ChatGPT said:**

It looks like **PM2** is not installed on your system. Let's install it and proceed.

**Step 1: Install PM2 Globally**

Run the following command to install **PM2**:

bash

CopyEdit

sudo npm install -g pm2

After installation, verify that PM2 is installed:

bash

CopyEdit

pm2 -v

If PM2 is installed correctly, it will return a version number.

**Step 2: Start Your Node.js Application**

Now, start your app using PM2:

bash

CopyEdit

pm2 start index.js --name "node-js-app"

Save the PM2 process list:

bash

CopyEdit

pm2 save

Check the status:

bash

CopyEdit

pm2 status

**Step 3: Enable Auto-Start on Server Reboot**

To make sure your app runs even after a system restart, run:

bash

CopyEdit

pm2 startup

This will return a command similar to:

bash

CopyEdit

sudo env PATH=$PATH:/usr/bin pm2 startup systemd -u ubuntu --hp /home/ubuntu

Copy and paste that command into the terminal and execute it.

**Step 4: Verify Your App is Running**

Check if your app is running:

bash

CopyEdit

curl http://localhost:3000

If you get the expected response, your app is running successfully! 🎉

Let me know if you face any issues. 🚀

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ pm2 start index.js --name "node-js-app" [PM2] Starting /home/ubuntu/node-js-app/index.js in fork\_mode (1 instance) [PM2] Done. ┌────┬────────────────┬─────────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐ │ id │ name │ namespace │ version │ mode │ pid │ uptime │ ↺ │ status │ cpu │ mem │ user │ watching │ ├────┼────────────────┼─────────────┼─────────┼─────────┼──────────┼────────┼──────┼───────────┼──────────┼──────────┼──────────┼──────────┤ │ 0 │ node-js-app │ default │ 1.0.0 │ fork │ 8993 │ 0s │ 0 │ online │ 0% │ 13.8mb │ ubuntu │ disabled │ └────┴────────────────┴─────────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘ ubuntu@ip-172-31-13-138:~/node-js-app$

**ChatGPT said:**

Your Node.js application is now successfully running with **PM2**! 🎉

**Step 5: Check If Your App is Running**

Run the following command:

bash

CopyEdit

curl http://localhost:3000

If the response is:

csharp

CopyEdit

Hello, World! This is a simple Node.js app by Deepanshu Kumar.

Then your app is working fine. ✅

**Step 6: Allow External Access to Your App**

Your app is currently accessible only inside the server. To make it publicly accessible, follow these steps:

**A. Allow Traffic on Port 3000**

Run:

bash

CopyEdit

sudo ufw allow 3000/tcp

If you are using **AWS Security Groups**, go to the **EC2 Dashboard** → **Security Groups** → Find your instance's security group → **Edit inbound rules** → **Add a rule:**

* **Type:** Custom TCP
* **Port Range:** 3000
* **Source:** Anywhere (0.0.0.0/0)

**B. Check Public Access**

Find your **Public IP**:

bash

CopyEdit

curl ifconfig.me

Now, visit:

cpp

CopyEdit

http://<your-public-ip>:3000

You should see your Node.js app running! 🚀

**Step 7: Ensure PM2 Runs After Reboot**

Run:

bash

CopyEdit

pm2 startup

It will return a command like:

bash

CopyEdit

sudo env PATH=$PATH:/usr/bin pm2 startup systemd -u ubuntu --hp /home/ubuntu

Copy and run that command.

Then, save the PM2 process list:

bash

CopyEdit

pm2 save

Now, even if the server reboots, your app will restart automatically.

**Step 8: Verify Deployment**

Test again:

bash

CopyEdit

curl http://<your-public-ip>:3000

If you see the expected response, your app is now fully deployed on AWS EC2! 🎉🚀

Let me know if you need any help! 😊

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ curl http://localhost:3000 Hello, World! This is a simple Node.js app by Deepanshu Kumar.

**ChatGPT said:**

Your Node.js app is running successfully on **localhost**! 🎉✅

Now, to make it accessible from the **internet**, follow these steps:

**Step 1: Find Your Public IP**

Run:

bash

CopyEdit

curl ifconfig.me

It will return your **EC2 public IP** (e.g., 54.123.45.67).

**Step 2: Allow External Access to Port 3000**

**A. Enable Firewall Rule (UFW)**

Run:

bash

CopyEdit

sudo ufw allow 3000/tcp

**B. Update AWS Security Group (if needed)**

1. Go to **AWS Console → EC2 → Security Groups**.
2. Find your instance’s **security group**.
3. Click **Inbound Rules → Edit → Add Rule**:
   * **Type:** Custom TCP
   * **Port Range:** 3000
   * **Source:** Anywhere (0.0.0.0/0)

**Step 3: Test External Access**

Now, open your browser and visit:

cpp

CopyEdit

http://<your-public-ip>:3000

or use **cURL**:

bash

CopyEdit

curl http://<your-public-ip>:3000

If you see:

csharp

CopyEdit

Hello, World! This is a simple Node.js app by Deepanshu Kumar.

🚀 **Your app is now live!** ✅

**Step 4: Keep App Running After Reboot**

To make sure your app restarts automatically after a reboot, run:

bash

CopyEdit

pm2 startup

Follow the output instructions (it will give a sudo command to run). Then, save the PM2 process list:

bash

CopyEdit

pm2 save

Now, your app will restart automatically after a system reboot.

**Final Check**

Try accessing:

cpp

CopyEdit

http://<your-public-ip>:3000

If it's working, **your app is fully deployed on AWS EC2! 🎉🚀**

Let me know if you need help! 😊

4o

**You said:**

ubuntu@ip-172-31-13-138:~/node-js-app$ curl ifconfig.me 15.207.247.66ubuntu@ip-172-31-13-138:~/curl http://^C000ic-ip>:3000 ubuntu@ip-172

**ChatGPT said:**

Your public IP is **15.207.247.66**. Now, try accessing your app from a browser or another system using:

cpp

CopyEdit

http://15.207.247.66:3000

or run:

bash

CopyEdit

curl http://15.207.247.66:3000

If it's not working, follow these troubleshooting steps:

**1️⃣ Check if Node.js App is Running**

Run:

bash

CopyEdit

pm2 list

If the status is **stopped**, restart it:

bash

CopyEdit

pm2 restart node-js-app

**2️⃣ Open Port 3000 in Security Groups**

If you are using AWS EC2, update your **Security Group** settings:

1. Go to **AWS Console → EC2 → Security Groups**.
2. Find the **security group** attached to your instance.
3. Click **Inbound Rules → Edit → Add Rule**:
   * **Type:** Custom TCP
   * **Port Range:** 3000
   * **Source:** 0.0.0.0/0 (for public access)

**3️⃣ Allow Port 3000 in UFW (Ubuntu Firewall)**

Run:

bash

CopyEdit

sudo ufw allow 3000/tcp

sudo ufw reload

**4️⃣ Restart the Server**

bash

CopyEdit

pm2 restart node-js-app

Then, try:

bash

CopyEdit

curl http://15.207.247.66:3000

Let me know if you need further help! 😊🚀

4o

Top of Form



O

Search

Reason

Bottom of Form

ChatGPT can make mistakes. Check i