

## Executing Nodejs Using Docker

Open windows command prompt & Create a folder.

Name of the folder is dockernodejs

```
C:\Users\P c>e:  
E:\>mkdir dockernodejs  
E:\>cd dockernodejs
```

Create a file named with app.js in the above folder

**Note: if vim doesn't work in cmd prompt create in normal Notepad.**

```
E:\dockernodejs>vim app.js
```

app.js

```
const express = require('express');  
const app = express();  
app.get('/',(req,res) => {  
    res.send("Docker Nodejs Example");  
});  
app.listen(3111,() => {  
    console.log("Server is running on port 3111")  
});
```

```
Command Prompt - vim app.js  
const express = require('express');  
const app = express();  
app.get('/',(req,res) => {  
    res.send("Docker Nodejs Example");  
});  
app.listen(3111,() => {  
    console.log("Server is running on port 3111")  
});  
~
```

```
npm init -y
```

```
npm init

E:\dockernodejs>npm init -y
wrote to E:\dockernodejs\package.json:

{
  "name": "dockernodejs",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
    "test": "echo \\" Error: no test specified \\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

```
npm i express
```

```
Command Prompt

E:\dockernodejs>npm i express
added 65 packages, and audited 66 packages in 8s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

E:\dockernodejs>
```

## Open git bash

```
MINGW64:/e/dockernodejs

P c@DESKTOP-TEKQI08 MINGW64 ~ (master)
$ cd e:

P c@DESKTOP-TEKQI08 MINGW64 /e
$ cd dockernodejs

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs
$ ls
app.js  node_modules/  package-lock.json  package.json

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs
$ echo "node_modules/" >> .gitignore
```

```
MINGW64:/e/dockernodejs

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs
$ git init
Initialized empty Git repository in E:/dockernodejs/.git/

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    app.js
    package-lock.json
    package.json

nothing added to commit but untracked files present (use "git add" to track)
```

```
MINGW64:/e/dockernodejs
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git config user.email "gujjuladeepak@gmail.com"
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git config user.name "Deepak"
```

```
MINGW64:/e/dockernodejs
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ ls
app.js  node_modules/  package-lock.json  package.json
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ touch Dockerfile
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ vim Dockerfile|
```

```
# Use the official Node.js image from the Docker Hub
FROM node:14
```

```
# Set the working directory inside the container
WORKDIR /usr/src/app
```

```
# Copy package.json and package-lock.json (if available)
COPY package*.json ./
```

```
# Install dependencies
RUN npm install
```

```
# Copy the rest of your application code
COPY ..
```

```
# Expose the port the app runs on
EXPOSE 3111
```

```
# Command to run the application
CMD ["node", "app.js"]
```

```
MINGW64:/e/dockernodejs
# Use the official Node.js image from the Docker Hub
FROM node:14

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

# Copy package.json and package-lock.json (if available)
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of your application code
COPY . .

# Expose the port the app runs on
EXPOSE 3111

# Command to run the application
CMD ["node", "app.js"]
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next
time Git touches it
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next
time Git touches it
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time
Git touches it
warning: in the working copy of 'package-lock.json', LF will be replaced by CRLF th
e next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the nex
t time Git touches it
```

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git commit -m "Both dockerfile and nodejs file"
[master (root-commit) 22abc07] Both dockerfile and nodejs file
 6 files changed, 818 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .init.swp
 create mode 100644 Dockerfile
 create mode 100644 app.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
```

\*\*\* Here you may not see .init.swp, without this also the

# program will execute

## Open github in any browser

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

/ dockernodejsexample

dockernodejsexample is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-garbanzo](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.

Active  
Go to s

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

dockernodejsexample

dockernodejsexample is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-garbanzo](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: [None](#) ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: [None](#) ▾

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

You are creating a public repository in your personal account.

[Create repository](#)

The screenshot shows a GitHub repository page for 'dockernodejsexample'. The page includes a search bar at the top, navigation links like Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings, and a header with a star count of 0 and a fork count of 0. A red box highlights the URL 'https://github.com/deepak574/dockernodejsexample.git' in the address bar. A red arrow points from the bottom right towards the URL bar.

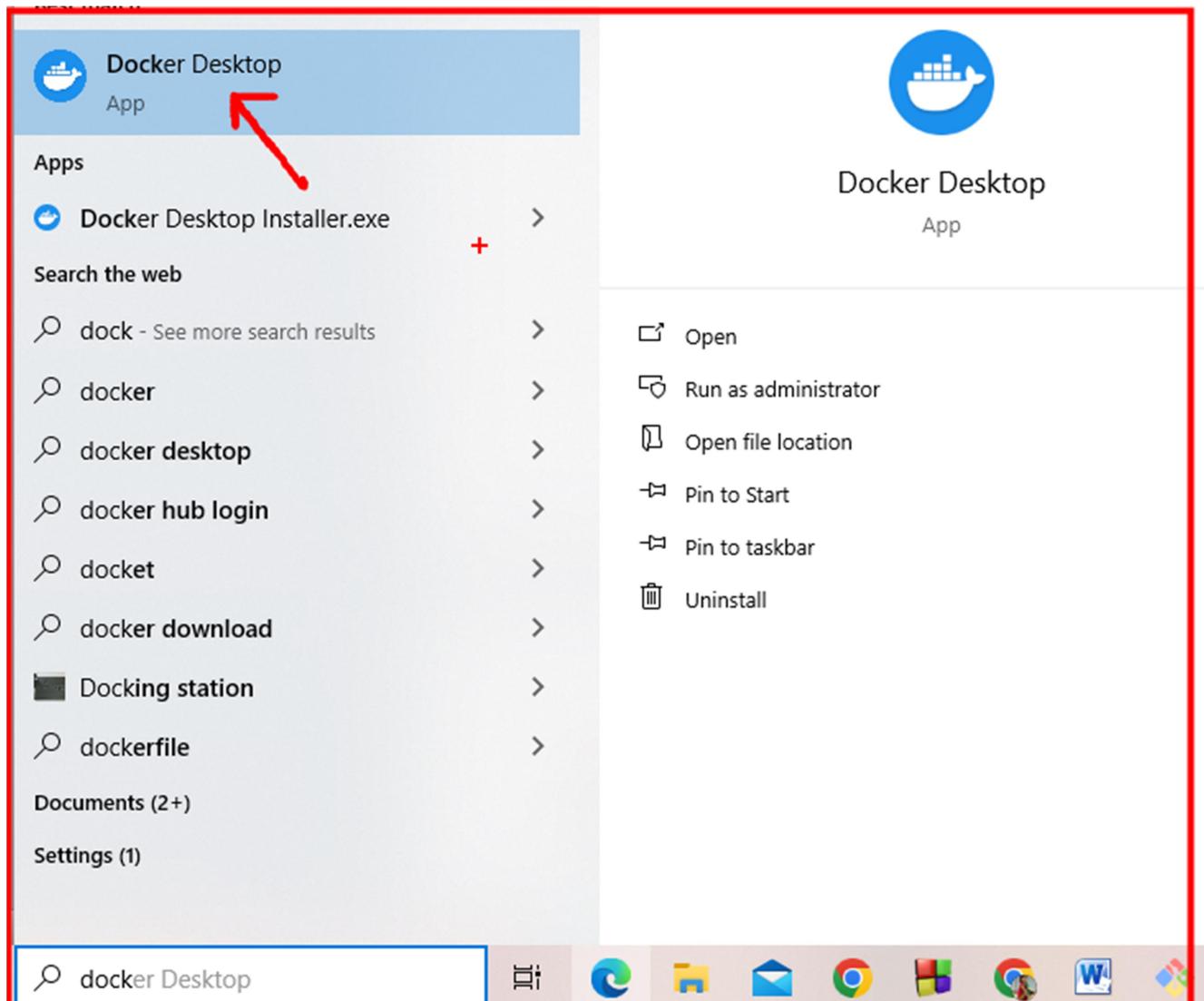
The screenshot shows a terminal window titled 'MINGW64:/e/dockernodejs'. It displays the following command history:

```
P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git remote add origin https://github.com/deepak574/dockernodejsexample.git

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (master)
$ git branch -M main

P c@DESKTOP-TEKQI08 MINGW64 /e/dockernodejs (main)
$ git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 8.36 KiB | 1.39 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/deepak574/dockernodejsexample.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

**Open Docker Desktop, after docker engine starts ,open cmd and execute the following commands**



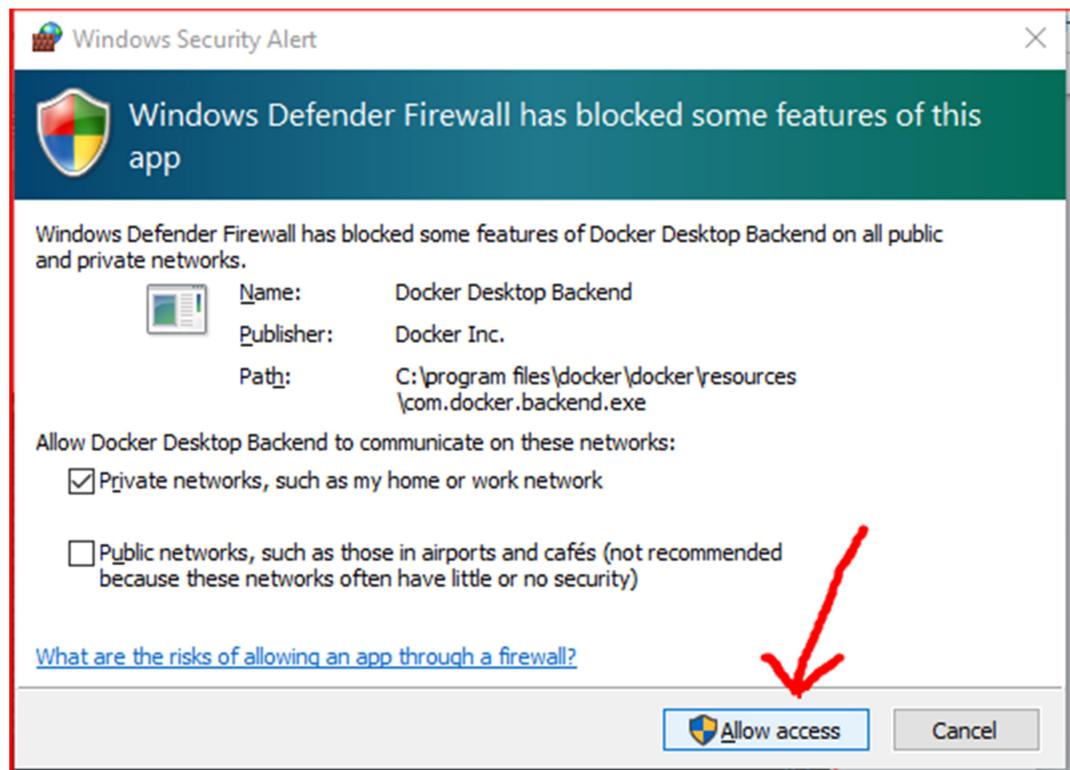
**docker build -t dockernodeja https://github.com/deepak574/dockernodejsexample.git**

```
[dockernodejs] docker build -t dockernodeja https://github.com/deepak574/dockernodejsexample.git
[+] Building 156.9s (8/8) FINISHED
=> [internal] load git source https://github.com/deepak574/dockernodejsexample.git
=> [internal] load metadata for docker.io/library/node:14
=> [1/5] FROM docker.io/library/node:14@sha256:a158db9b4e3fa813faacbc59eb8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> => resolve docker.io/library/node:14@sha256:a158db9b4e3fa813faacbc59eb8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> sha256:bd27a8e801329007574c756fb9a4646e820d28e964e873de352603f7224ceb 450B / 450B
=> sha256:08c8cc2f24adcb64e602e086fc9446b0a541e8acd9ad72d2e90df2ba22f158b3 2.99MB / 2.99MB
=> sha256:5f32edc5f278edda4fc571c880b527735a29ae8752b52cd865f058378a99e 35.24MB / 35.24MB
=> sha256:6f51ee005deac0d9989841ba0b03f613aec6bbc9b83d8 4.19KB / 4.19KB
=> sha256:0d948df5894511ce28a95e2925a75e8a44bd0634c39ad734fdftbae23d1b1569 191.85MB / 191.85MB
=> sha256:1de76e268b103d05fa8950e0f77951ff454b12b63429c245fd6adfd09f5f9ee2 51.88MB / 51.88MB
=> sha256:3d2201bd9951ccf12851a598820de03d34a1701d1ccb9ac9fd3a5bc952cb131 10.00MB / 10.00MB
=> sha256:b253aeaefea7e0e71b60008df01de101a38a045ff7bc56e3b6bf7c705cc5 7.86MB / 7.86MB
=> sha256:2ff1d74c1c74a25258bf6fb8adbe727f84518f5f5765ca845ebc747976c408 58.45MB / 58.45MB
=> extracting sha256_2ff1d74c1c74a25258bf6fb8adbe727f84518f5f5765ca845ebc747976c408
=> extracting sha256_b253aeaefea7e0e71b60008df01de101a38a045ff7bc656e3b6bf7c705cc5
=> extracting sha256_d32201bd9951ccf12851a598820de03d34a17011dccb9ac9fd3a5bc952cb131
=> extracting sha256_0d948df5894511ce28a95e2925a75e8a44bd0634c39ad734fdftbae23d1b1569
=> extracting sha256_6f51ee005deac0d9989841b8c606ebf250ebef1a1a0b03f613aec6bbc9b83d8
=> extracting sha256_5f32edc5f278edda4fc571c880b5277355a29ae8ff52b52cd865f058378a590
=> extracting sha256_0d27a8e861329007574c6766fb9a46d48e20d2c8e964e873de352603f22c4ceb
=> extracting sha256_naming to docker.io/library/dockernodeja:latest
=> unpacking to docker.io/library/dockernodeja:latest
[2/5] WORKDIR /usr/src/app
[3/5] COPY package.json .
[4/5] RUN npm install
[5/5] COPY .
=> exporting to image
=> exporting layers
=> exporting manifest sha256:7bf1a43c3170e6a8074463bd3a0fc199a81ce2743825d08fd70f0f7ed63a0e0
=> exporting config sha256:4f02e816ec7790c084173bb4533be415f5f67bbac248114dc48eb013821b49b
=> exporting attestation manifest sha256:01d7d785bdb913ec6e93376a395cd85e88c931eec4b7da5076835e9619abf3
=> exporting manifest list sha256:3f4760d10597195062b3a9cdceefcd1cb60c138c9afc697e9d60a39d0ee6bf
=> naming to docker.io/library/dockernodeja:latest
=> unpacking to docker.io/library/dockernodeja:latest
What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Activate Windows  
Go to Settings to activate Windows.

**docker run -d -p 3111:3111 dockernodeja**

```
E:\dockernodejs>docker run -d -p 3111:3111 dockernodeja
cf147870e08908de25cc16203d4565ae2e313b57a02e201157ff599daf9745ae
```



**Open any browser and type localhost:3111**

