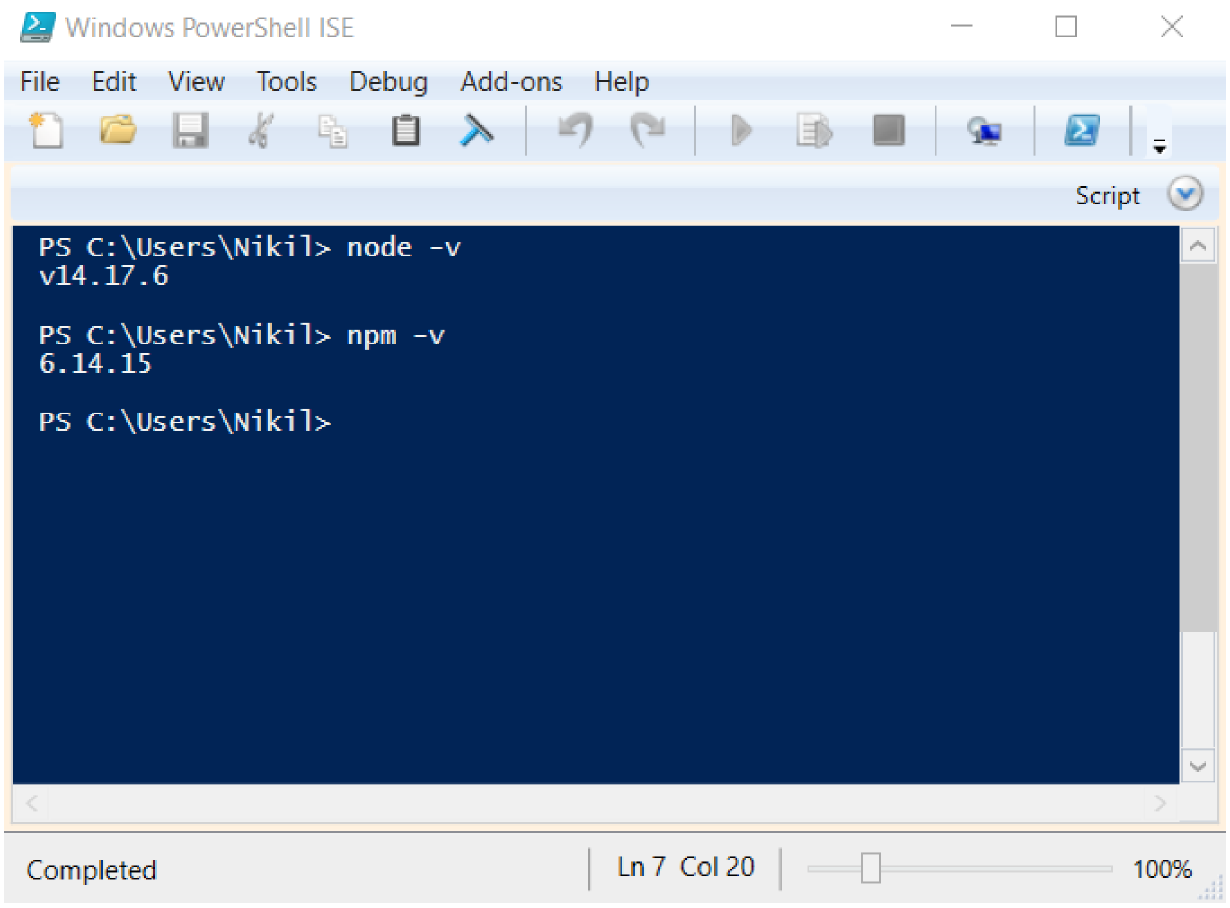
Team - 03

Nikhil S A (18BEC033)
Jagadeesh C (18BCS033)
Arun Kumar S M (18BCS013)
Nithin R (18BEC034)
Kiran D (18BEC022)
Jeevan R H (18BEC017)

Assignment-1:

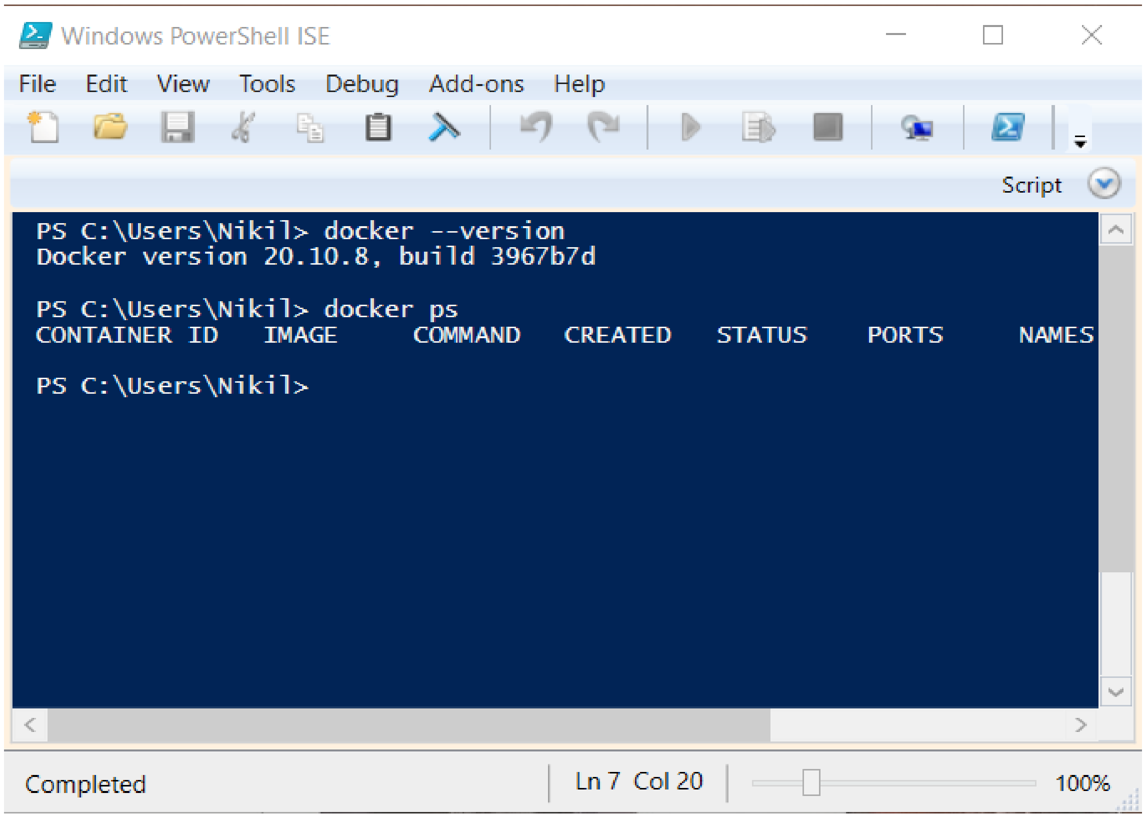
Developing and deploying a Node.js app from Docker to Kubernetes

Step1: Install Node.js and npm.



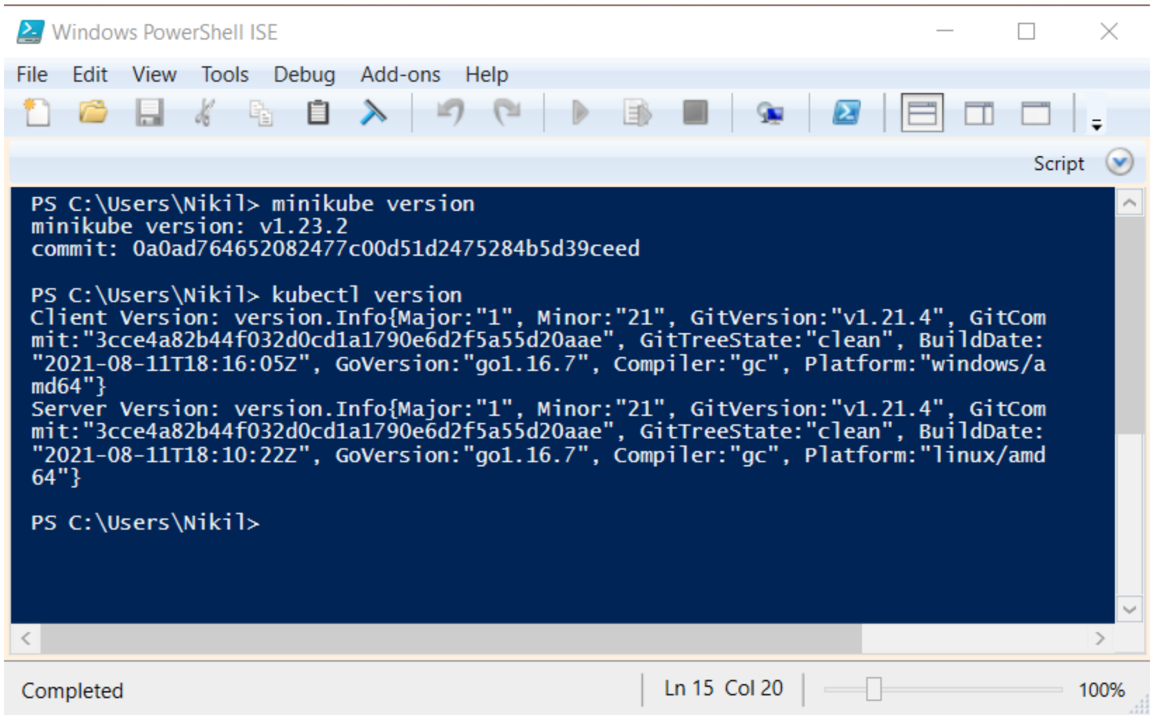
```
PS C:\Users\Nikil> node -v
v14.17.6

PS C:\Users\Nikil> npm -v
6.14.15

PS C:\Users\Nikil>
```

Step2: Installing Docker.



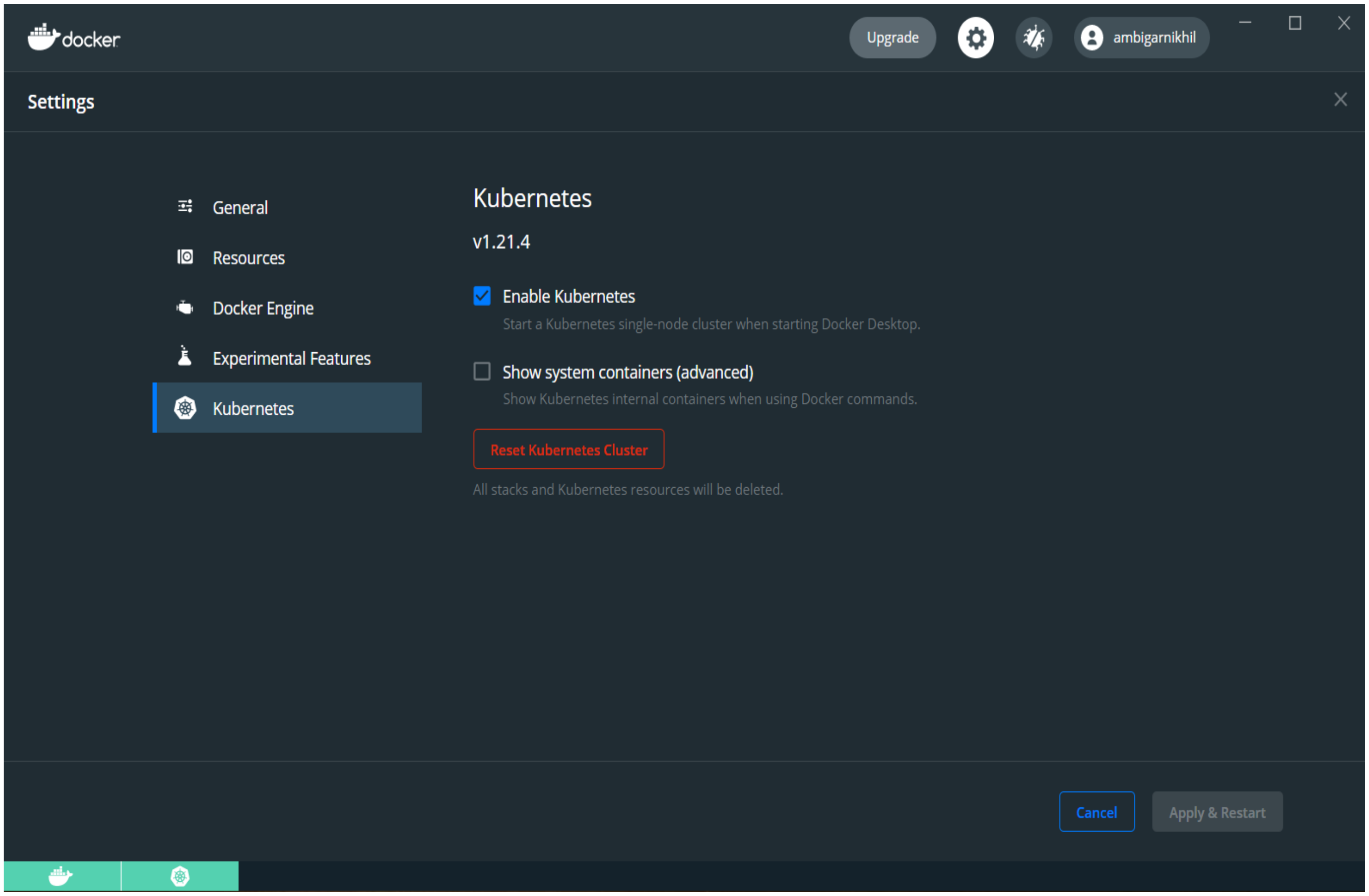
```
PS C:\Users\Nikil> docker --version
Docker version 20.10.8, build 3967b7d

PS C:\Users\Nikil> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

PS C:\Users\Nikil>
```

Step3: Minikube and kubectl installation.



```
PS C:\Users\Nikil> minikube version
minikube version: v1.23.2
commit: 0a0ad764652082477c00d51d2475284b5d39ceed

PS C:\Users\Nikil> kubectl version
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.4", GitCom
mit:"3cce4a82b44f032d0cd1a1790e6d2f5a55d20aae", GitTreeState:"clean", BuildDate:
"2021-08-11T18:16:05Z", GoVersion:"go1.16.7", Compiler:"gc", Platform:"windows/a
md64"}
Server Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.4", GitCom
mit:"3cce4a82b44f032d0cd1a1790e6d2f5a55d20aae", GitTreeState:"clean", BuildDate:
"2021-08-11T18:10:22Z", GoVersion:"go1.16.7", Compiler:"gc", Platform:"linux/amd
64"}

PS C:\Users\Nikil>
```

Step4: Enable Kubernetes service with docker.



Kubernetes

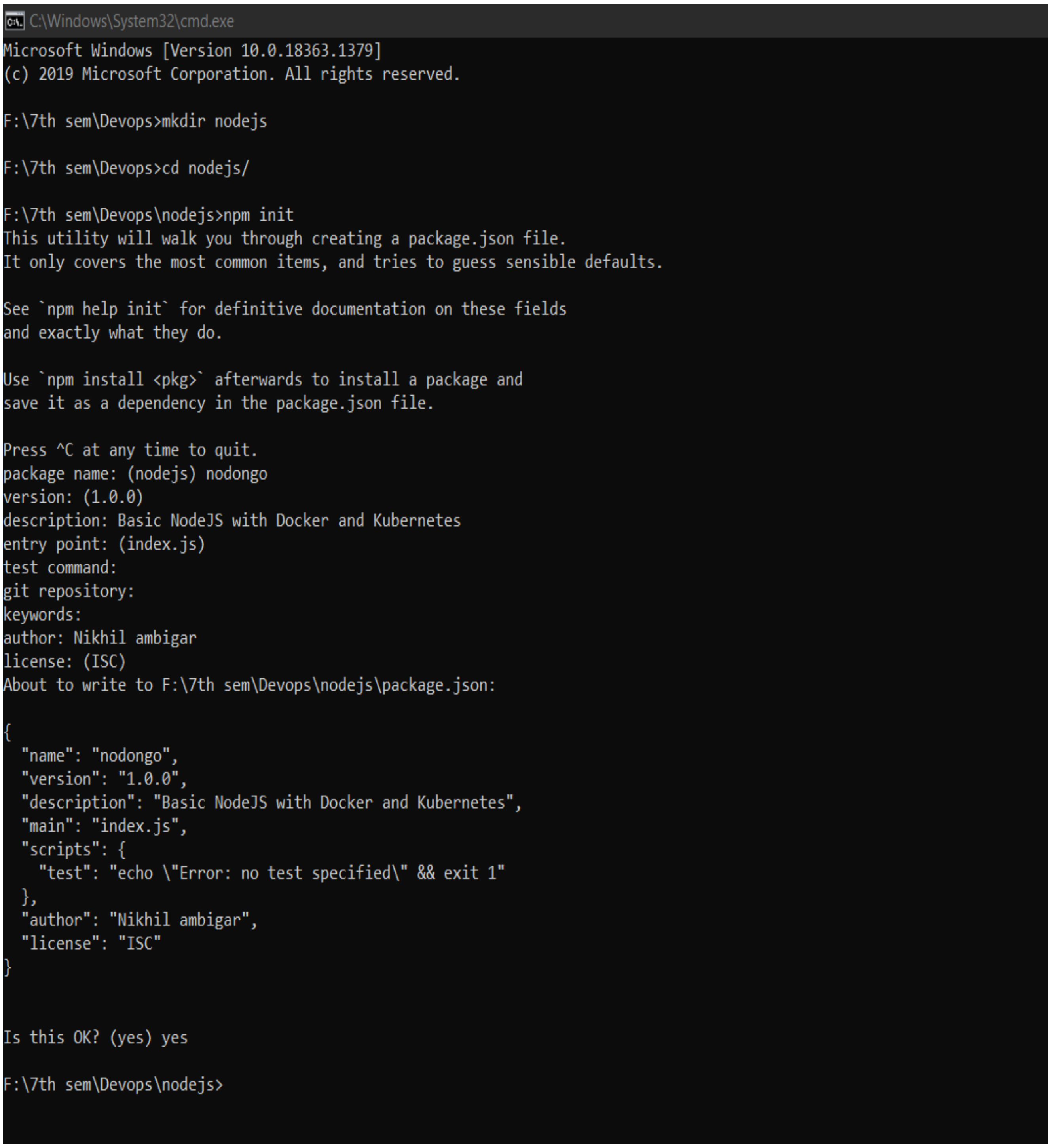
v1.21.4

[✓] Enable Kubernetes
Start a Kubernetes single-node cluster when starting Docker Desktop.

[ ] Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.

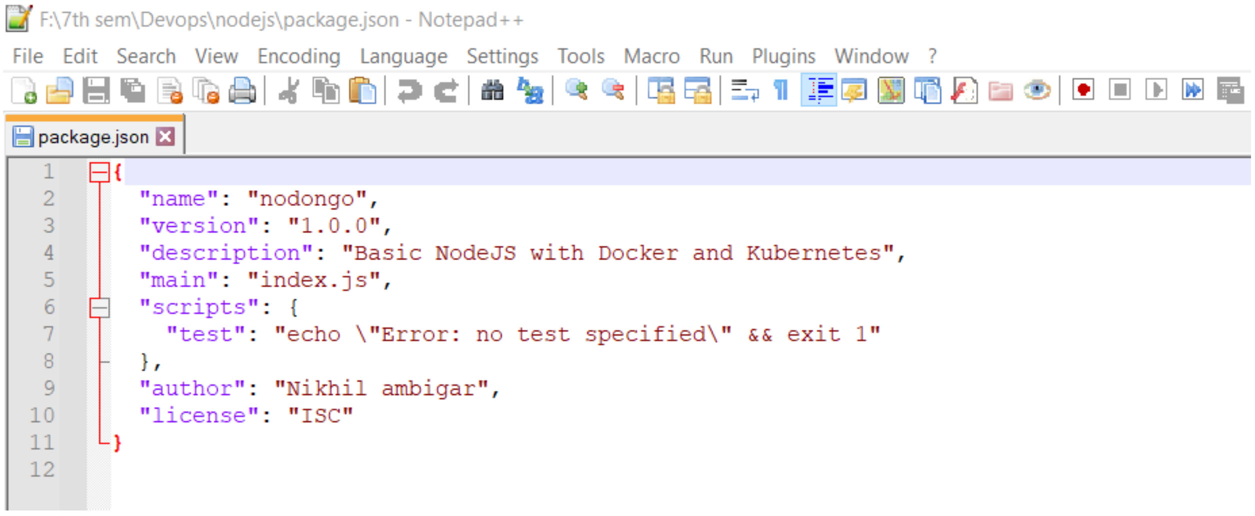
All stacks and Kubernetes resources will be deleted.

Step5: Make A Separate Directory And Initialize The Node Application.



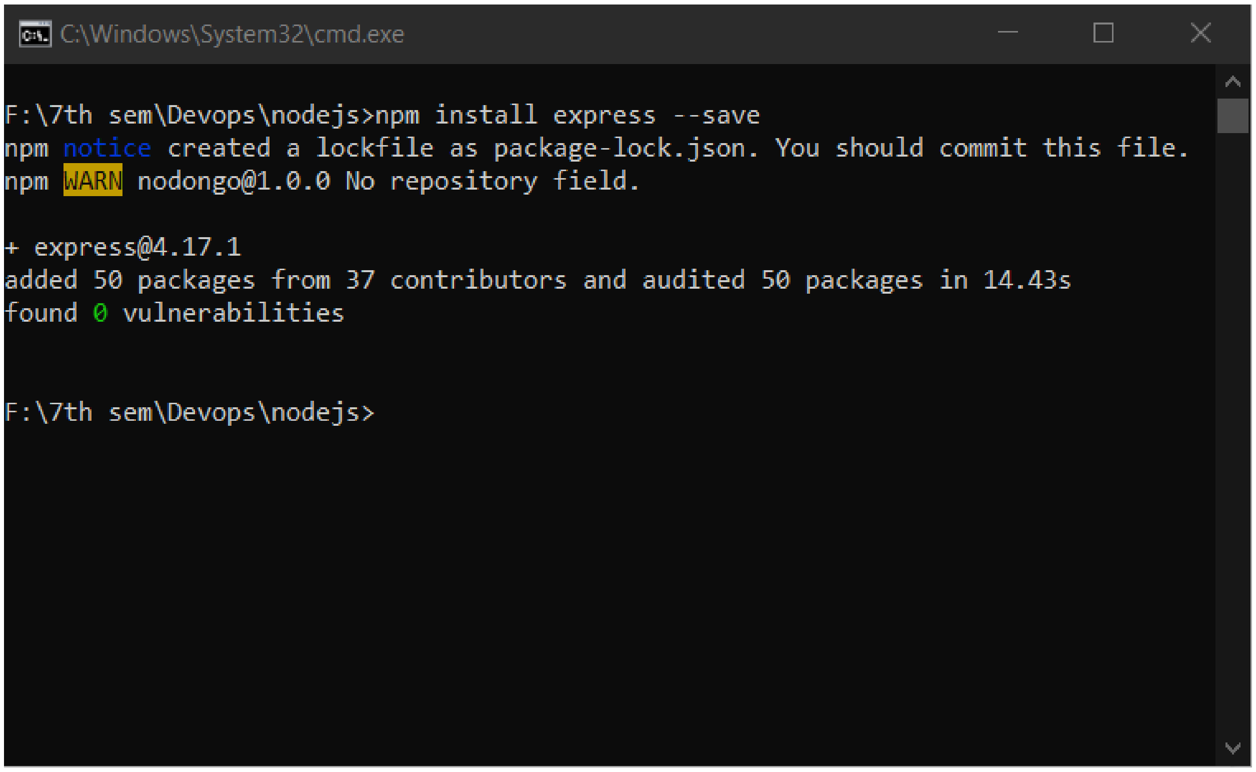
```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\7th sem\Devops>mkdir nodejs

F:\7th sem\Devops>cd nodejs/

F:\7th sem\Devops\nodejs>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs) nodongo
version: (1.0.0)
description: Basic NodeJS with Docker and Kubernetes
entry point: (index.js)
test command:
git repository:
keywords:
author: Nikhil ambigar
license: (ISC)
About to write to F:\7th sem\Devops\nodejs\package.json:

{
  "name": "nodongo",
  "version": "1.0.0",
  "description": "Basic NodeJS with Docker and Kubernetes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Nikhil ambigar",
  "license": "ISC"
}


Is this OK? (yes) yes

F:\7th sem\Devops\nodejs>
```

View of Package.json file :


F:\7th sem\Devops\nodejs\package.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package.json

```
{
  "name": "nodongo",
  "version": "1.0.0",
  "description": "Basic NodeJS with Docker and Kubernetes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Nikhil ambigar",
  "license": "ISC"
}
```

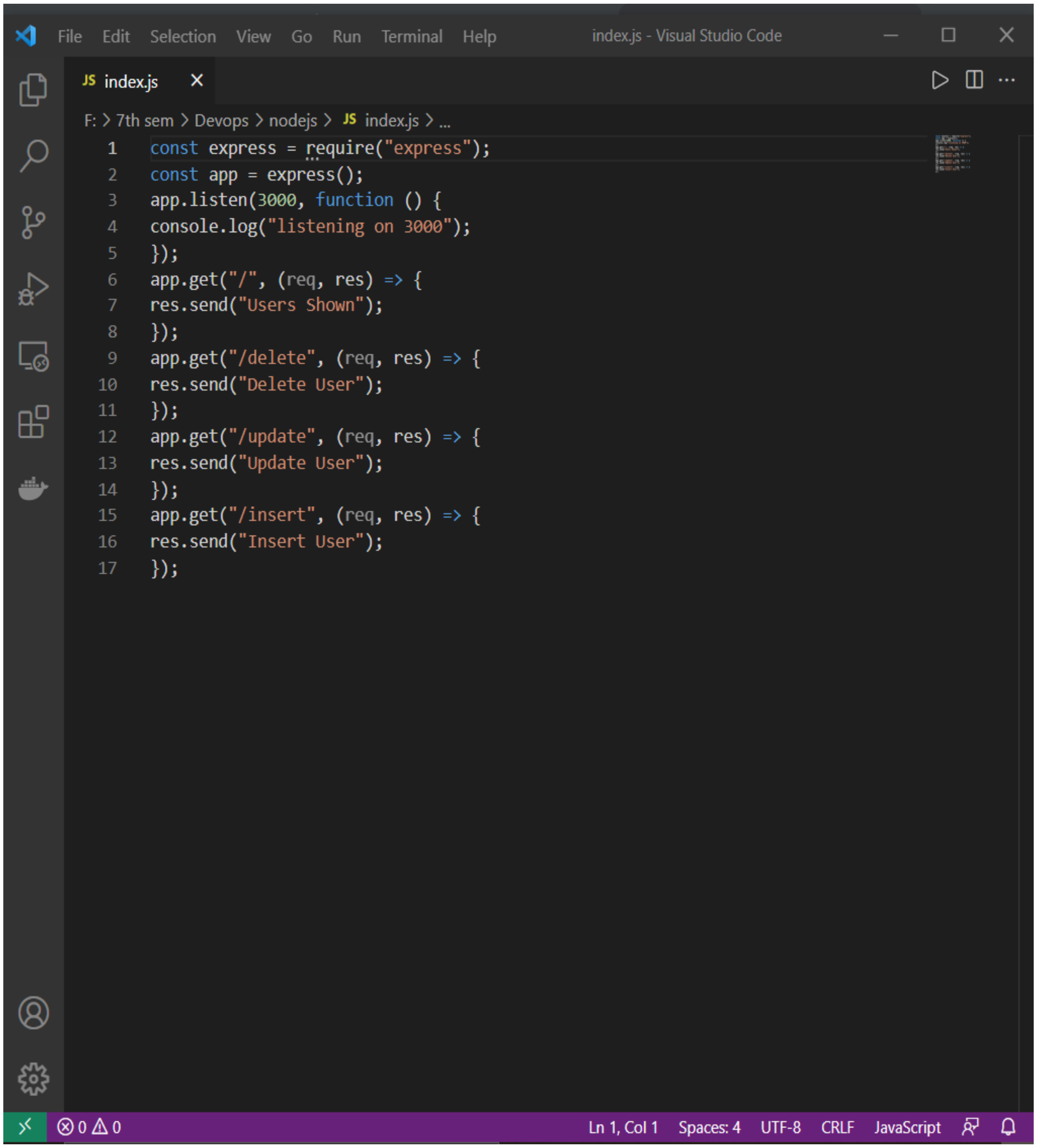
Step6: Installing Express.



```
F:\7th sem\Devops\nodejs>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN nodongo@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 14.43s
found 0 vulnerabilities


F:\7th sem\Devops\nodejs>
```

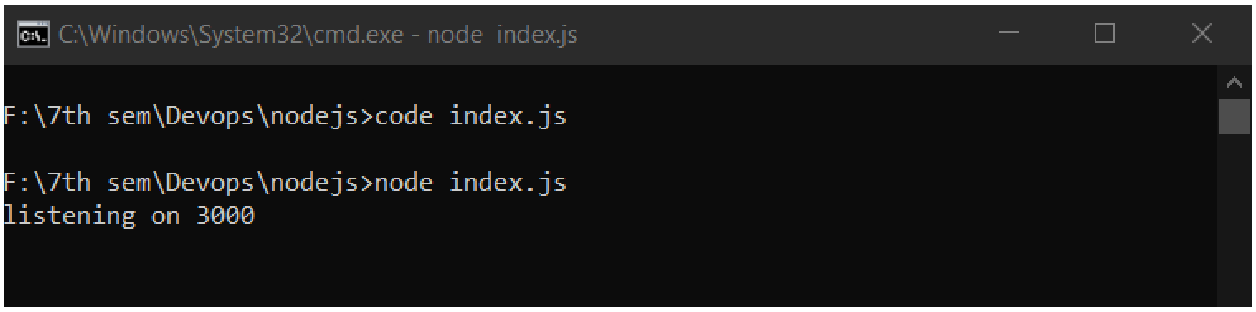
Step7: Make index.js file and write some code.

Command to run on terminal: code index.js (as we are using VScode as default editor).


File Edit Selection View Go Run Terminal Help
index.js - Visual Studio Code
index.js
F: > 7th sem > Devops > nodejs > index.js > ...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

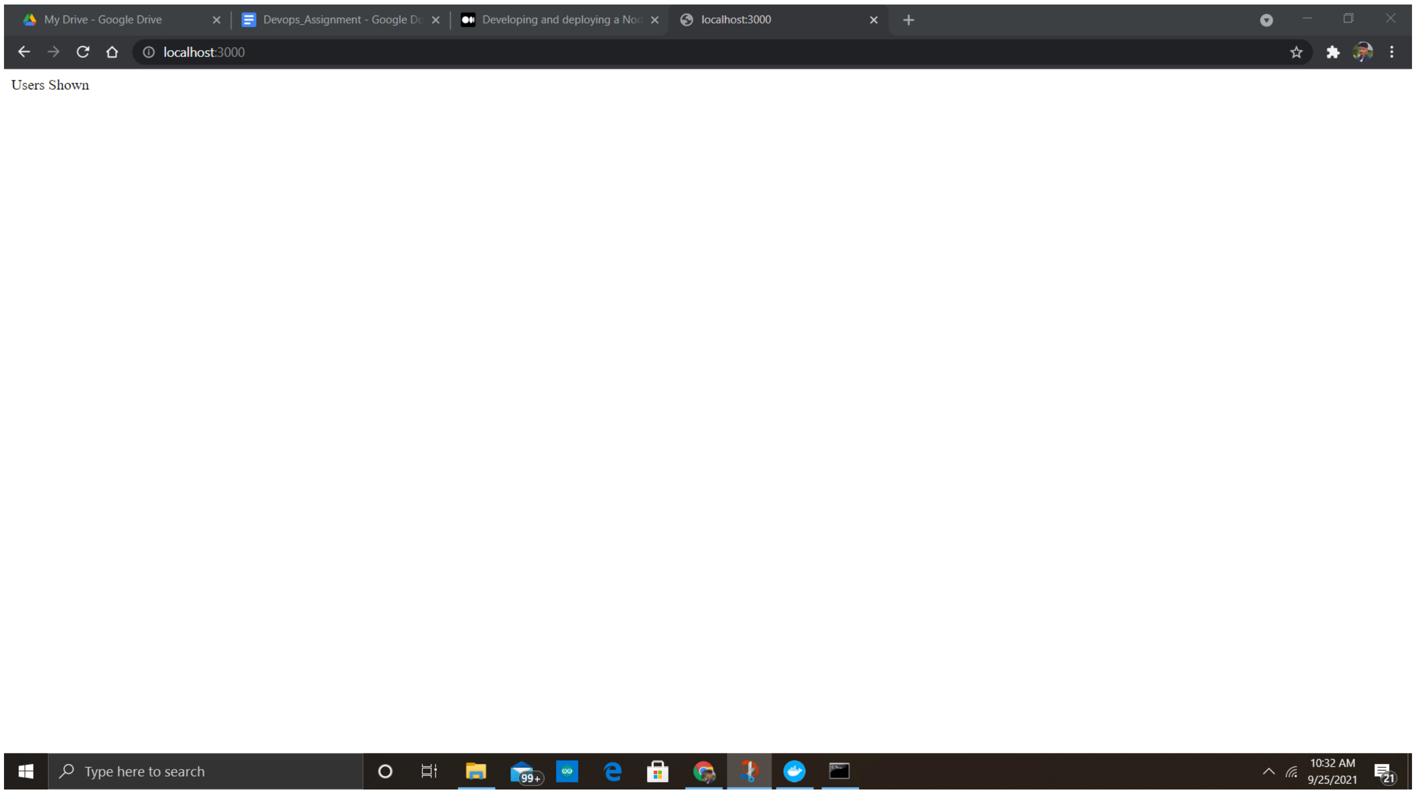
```
const express = require("express");
const app = express();
app.listen(3000, function () {
console.log("listening on 3000");
});
app.get("/", (req, res) => {
res.send("Users Shown");
});
app.get("/delete", (req, res) => {
res.send("Delete User");
});
app.get("/update", (req, res) => {
res.send("Update User");
});
app.get("/insert", (req, res) => {
res.send("Insert User");
});
```

After writing the code in the index.js file run the following command in terminal.
node index.js



```
C:\Windows\System32\cmd.exe - node  index.js

F:\7th sem\Devops\nodejs>code index.js

F:\7th sem\Devops\nodejs>node index.js
listening on 3000
```

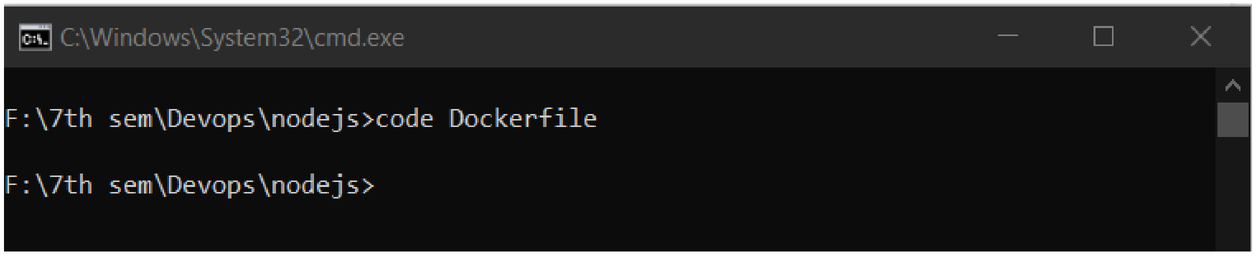
You can now check the server by using the following command, and browsing localhost:3000/



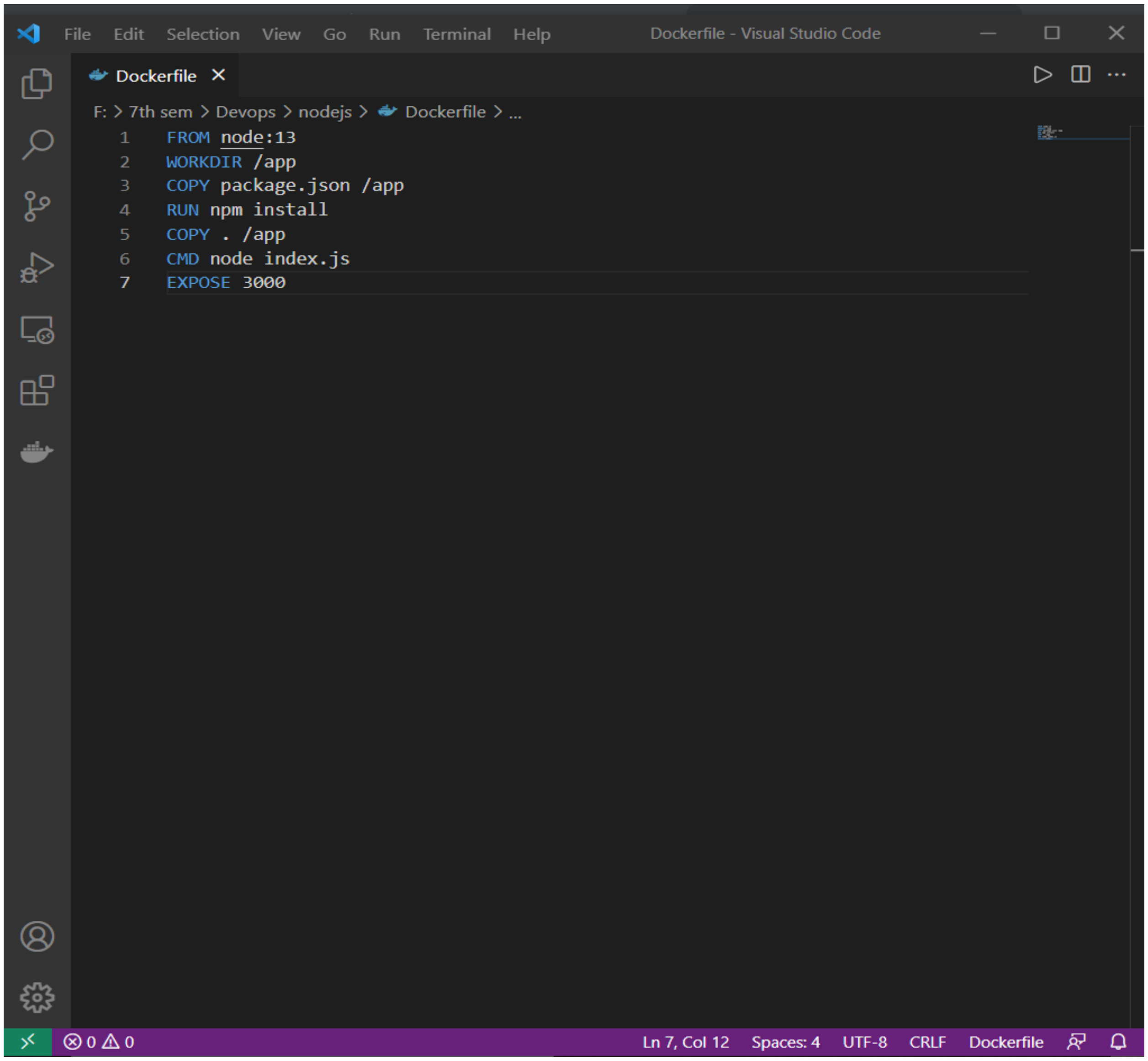
```
Users Shown
```

Step8: Dockerizing The Node Server.

For creating the Dockerfile run the following command on terminal:
code Dockerfile



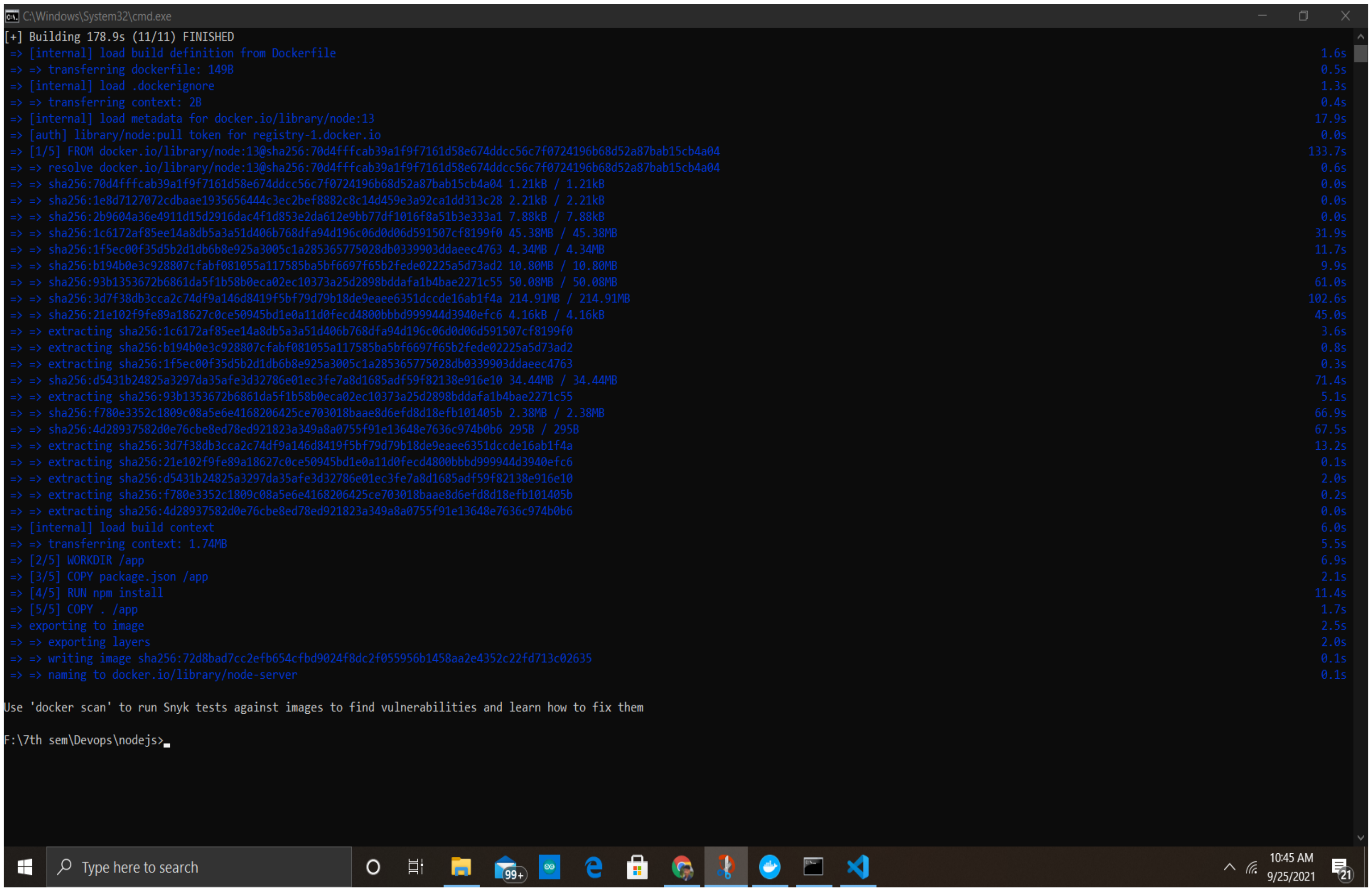
```
F:\7th sem\Devops\nodejs>code Dockerfile

F:\7th sem\Devops\nodejs>
```



```
FROM node:13
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
CMD node index.js
EXPOSE 3000
```

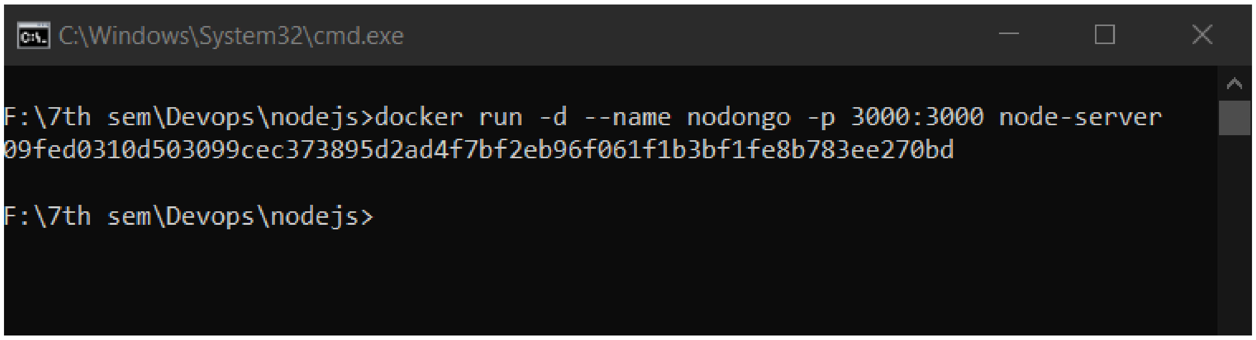
From Dockerfile we’ll start building our image by running the following command on terminal:

docker build -t node-server .



```
[+] Building 178.9s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                                    1.6s
 => => transferring dockerfile: 149B                                                                                    0.5s
 => [internal] load .dockerignore                                                                                       1.3s
 => => transferring context: 2B                                                                                         0.4s
 => [internal] load metadata for docker.io/library/node:13                                                             17.9s
 => [auth] library/node:pull token for registry-1.docker.io                                                             0.0s
 => [1/5] FROM docker.io/library/node:13@sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04      133.7s
 => => resolve docker.io/library/node:13@sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04        0.6s
 => => sha256:70d4fffcab39a1f9f7161d58e674ddcc56c7f0724196b68d52a87bab15cb4a04 1.21kB / 1.21kB                          0.0s
 => => sha256:1e8d7127072cdbaae1935656444c3ec2bef8882c8c14d459e3a92ca1dd313c28 2.21kB / 2.21kB                          0.0s
 => => sha256:2b9604a36e4911d15d2916dac4f1d853e2da612e9bb77df1016f8a51b3e333a1 7.88kB / 7.88kB                          0.0s
 => => sha256:1c6172af85ee14a8db5a3a51d406b768dfa94d196c06d0d06d591507cf8199f0 45.38MB / 45.38MB                       31.9s
 => => sha256:1f5ec00f35d5b2d1db6b8e925a3005c1a285365775028db0339903ddaeec4763 4.34MB / 4.34MB                         11.7s
 => => sha256:b194b0e3c928807cfabf081055a117585ba5bf6697f65b2fede02225a5d73ad2 10.80MB / 10.80MB                        9.9s
 => => sha256:93b1353672b6861da5f1b58b0eca02ec10373a25d2898bddafa1b4bae2271c55 50.08MB / 50.08MB                       61.0s
 => => sha256:3d7f38db3cca2c74df9a146d8419f5bf79d79b18de9eaee6351dccde16ab1f4a 214.91MB / 214.91MB                    102.6s
 => => sha256:21e102f9fe89a18627c0ce50945bd1e0a11d0fecd4800bbbd999944d3940efc6 4.16kB / 4.16kB                         45.0s
 => => extracting sha256:1c6172af85ee14a8db5a3a51d406b768dfa94d196c06d0d06d591507cf8199f0                               3.6s
 => => extracting sha256:b194b0e3c928807cfabf081055a117585ba5bf6697f65b2fede02225a5d73ad2                               0.8s
 => => extracting sha256:1f5ec00f35d5b2d1db6b8e925a3005c1a285365775028db0339903ddaeec4763                               0.3s
 => => sha256:d5431b24825a3297da35afe3d32786e01ec3fe7a8d1685adf59f82138e916e10 34.44MB / 34.44MB                       71.4s
 => => extracting sha256:93b1353672b6861da5f1b58b0eca02ec10373a25d2898bddafa1b4bae2271c55                               5.1s
 => => sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b 2.38MB / 2.38MB                         66.9s
 => => sha256:4d28937582d0e76cbe8ed78ed921823a349a8a0755f91e13648e7636c974b0b6 295B / 295B                             67.5s
 => => extracting sha256:3d7f38db3cca2c74df9a146d8419f5bf79d79b18de9eaee6351dccde16ab1f4a                              13.2s
 => => extracting sha256:21e102f9fe89a18627c0ce50945bd1e0a11d0fecd4800bbbd999944d3940efc6                               0.1s
 => => extracting sha256:d5431b24825a3297da35afe3d32786e01ec3fe7a8d1685adf59f82138e916e10                               2.0s
 => => extracting sha256:f780e3352c1809c08a5e6e4168206425ce703018baae8d6efd8d18efb101405b                               0.2s
 => => extracting sha256:4d28937582d0e76cbe8ed78ed921823a349a8a0755f91e13648e7636c974b0b6                               0.0s
 => [internal] load build context                                                                                       6.0s
 => => transferring context: 1.74MB                                                                                     5.5s
 => [2/5] WORKDIR /app                                                                                                  6.9s
 => [3/5] COPY package.json /app                                                                                        2.1s
 => [4/5] RUN npm install                                                                                              11.4s
 => [5/5] COPY . /app                                                                                                   1.7s
 => exporting to image                                                                                                  2.5s
 => => exporting layers                                                                                                 2.0s
 => => writing image sha256:72d8bad7cc2efb654cfbd9024f8dc2f055956b1458aa2e4352c22fd713c02635                            0.1s
 => => naming to docker.io/library/node-server                                                                          0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

F:\7th sem\Devops\nodejs>
```

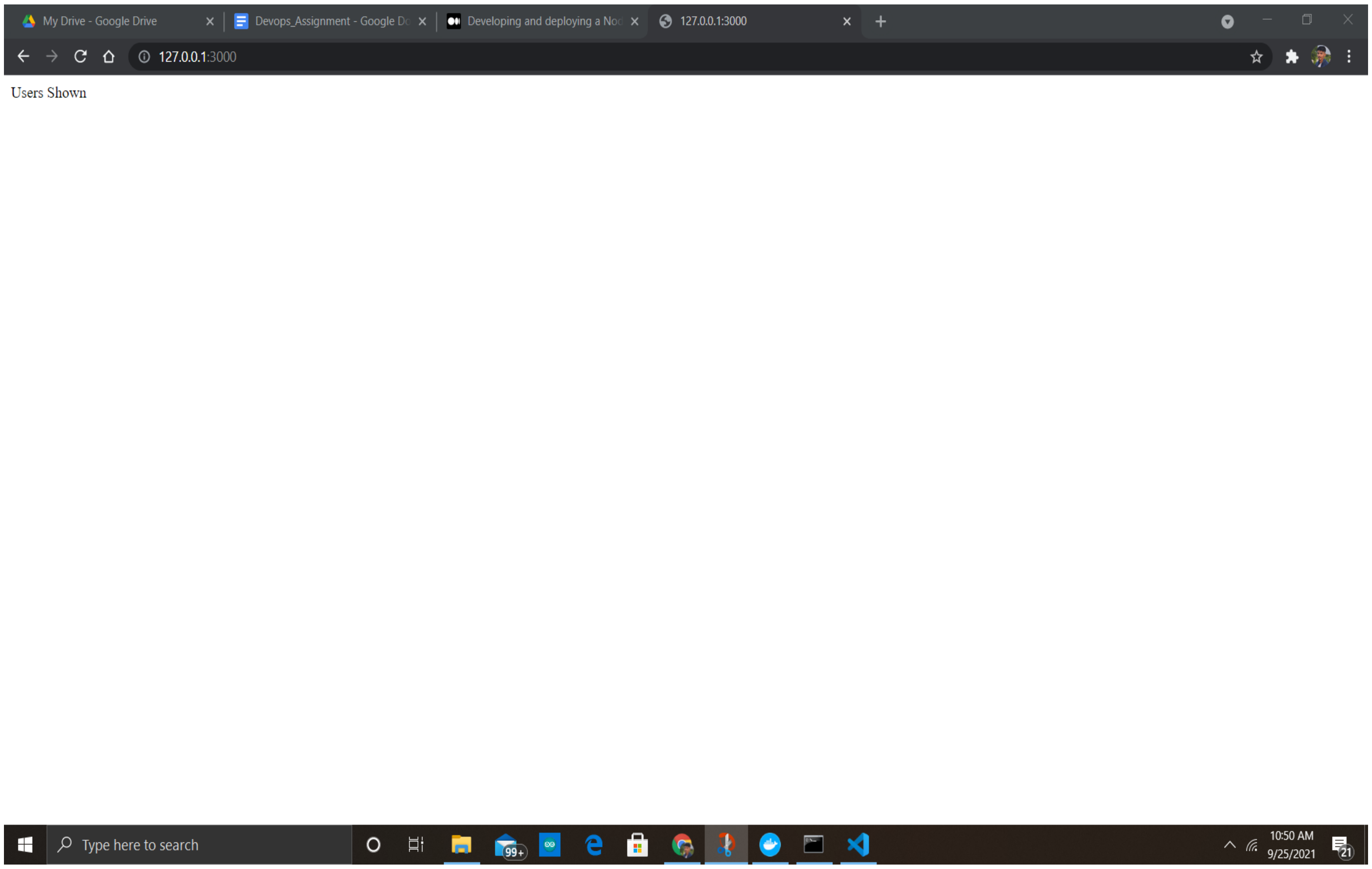
Step9: Create And Run The Container.



C:\Windows\System32\cmd.exe

```
F:\7th sem\Devops\nodejs>docker run -d --name nodongo -p 3000:3000 node-server
09fed0310d503099cec373895d2ad4f7bf2eb96f061f1b3bf1fe8b783ee270bd

F:\7th sem\Devops\nodejs>
```

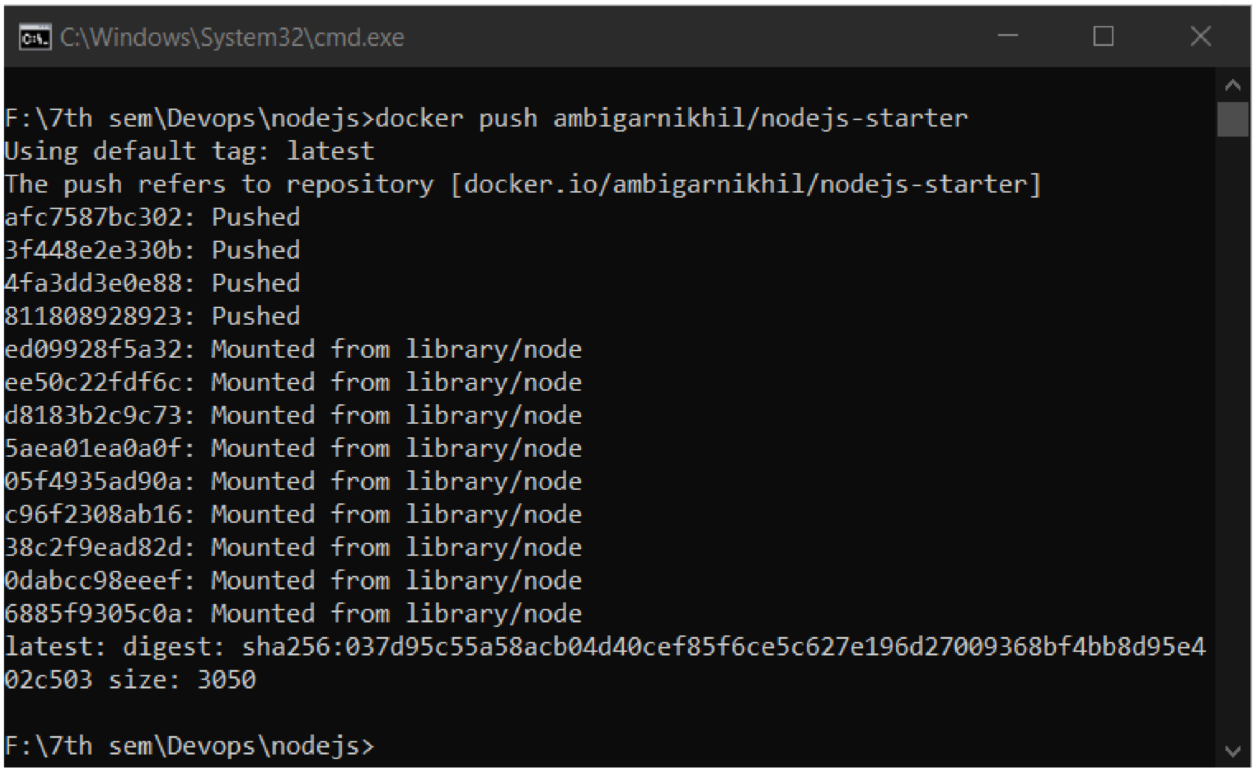
Go to the browser and browse the following address **127.0.0.1:3000** to test that it's running.



Users Shown

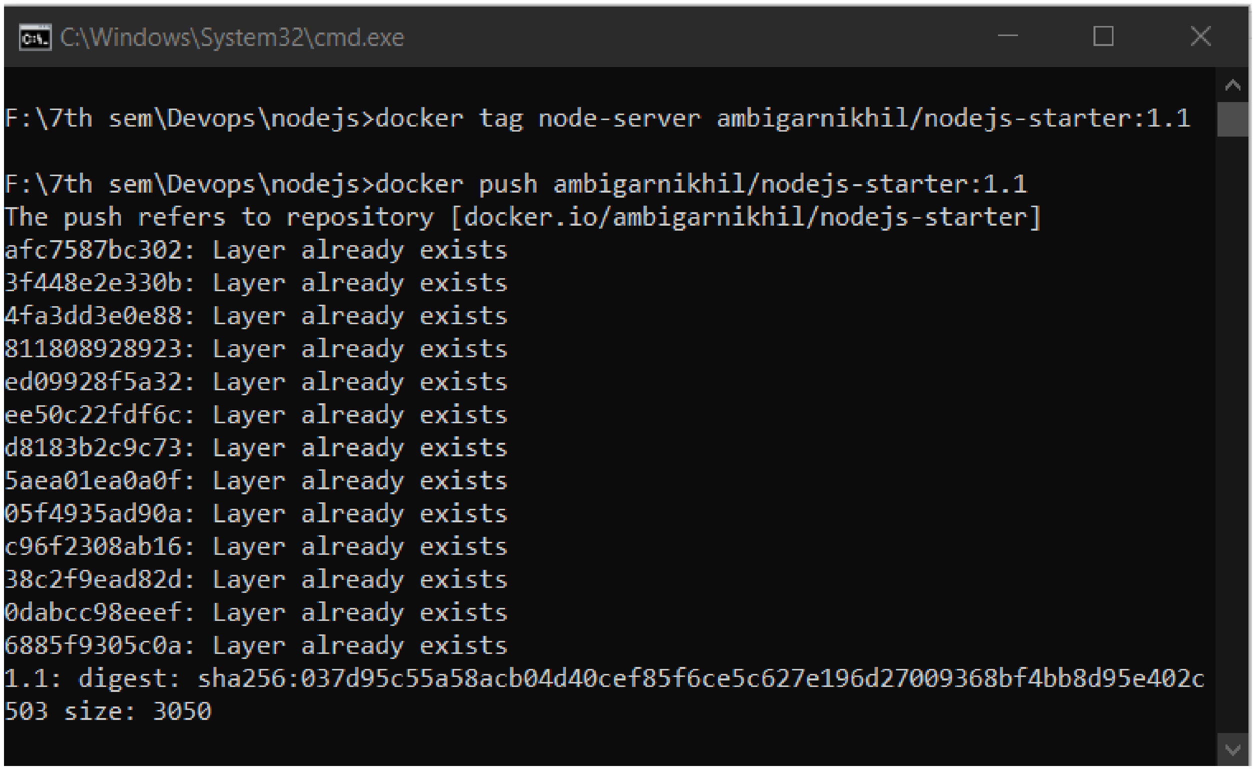
Step9: Upload The Image To Docker Registry Docker Hub.

Latest push:

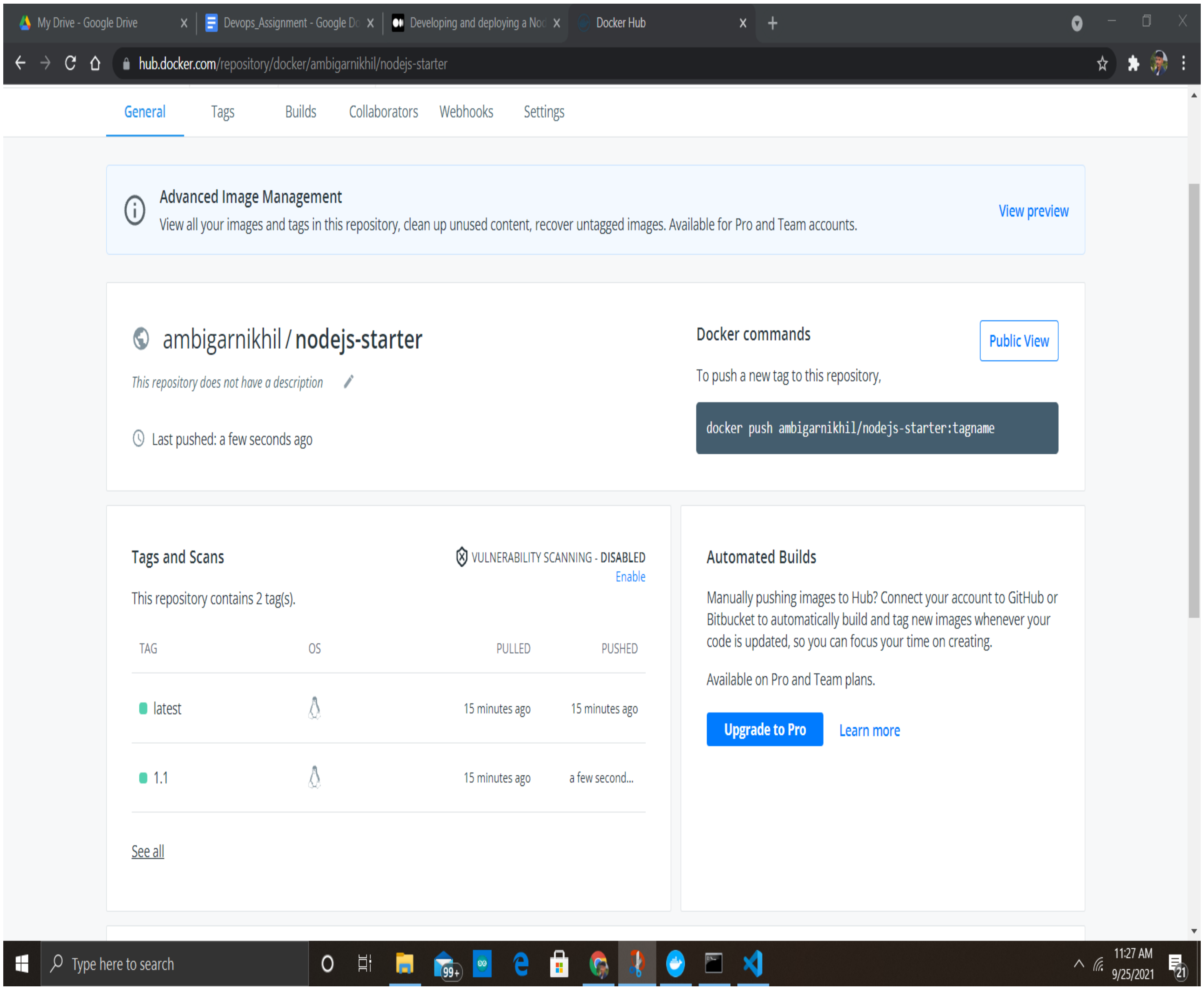

C:\Windows\System32\cmd.exe

```
F:\7th sem\Devops\nodejs>docker push ambigarnikhil/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/ambigarnikhil/nodejs-starter]
afc7587bc302: Pushed
3f448e2e330b: Pushed
4fa3dd3e0e88: Pushed
811808928923: Pushed
ed09928f5a32: Mounted from library/node
ee50c22fdf6c: Mounted from library/node
d8183b2c9c73: Mounted from library/node
5aea01ea0a0f: Mounted from library/node
05f4935ad90a: Mounted from library/node
c96f2308ab16: Mounted from library/node
38c2f9ead82d: Mounted from library/node
0dabcc98eeef: Mounted from library/node
6885f9305c0a: Mounted from library/node
latest: digest: sha256:037d95c55a58acb04d40cef85f6ce5c627e196d27009368bf4bb8d95e4
02c503 size: 3050

F:\7th sem\Devops\nodejs>
```

Version 1.1 push:


C:\Windows\System32\cmd.exe

```
F:\7th sem\Devops\nodejs>docker tag node-server ambigarnikhil/nodejs-starter:1.1

F:\7th sem\Devops\nodejs>docker push ambigarnikhil/nodejs-starter:1.1
The push refers to repository [docker.io/ambigarnikhil/nodejs-starter]
afc7587bc302: Layer already exists
3f448e2e330b: Layer already exists
4fa3dd3e0e88: Layer already exists
811808928923: Layer already exists
ed09928f5a32: Layer already exists
ee50c22fdf6c: Layer already exists
d8183b2c9c73: Layer already exists
5aea01ea0a0f: Layer already exists
05f4935ad90a: Layer already exists
c96f2308ab16: Layer already exists
38c2f9ead82d: Layer already exists
0dabcc98eeef: Layer already exists
6885f9305c0a: Layer already exists
1.1: digest: sha256:037d95c55a58acb04d40cef85f6ce5c627e196d27009368bf4bb8d95e402c
503 size: 3050
```



hub.docker.com/repository/docker/ambigarnikhil/nodejs-starter

Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

ambigarnikhil / nodejs-starter

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push ambigarnikhil/nodejs-starter:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 2 tag(s).

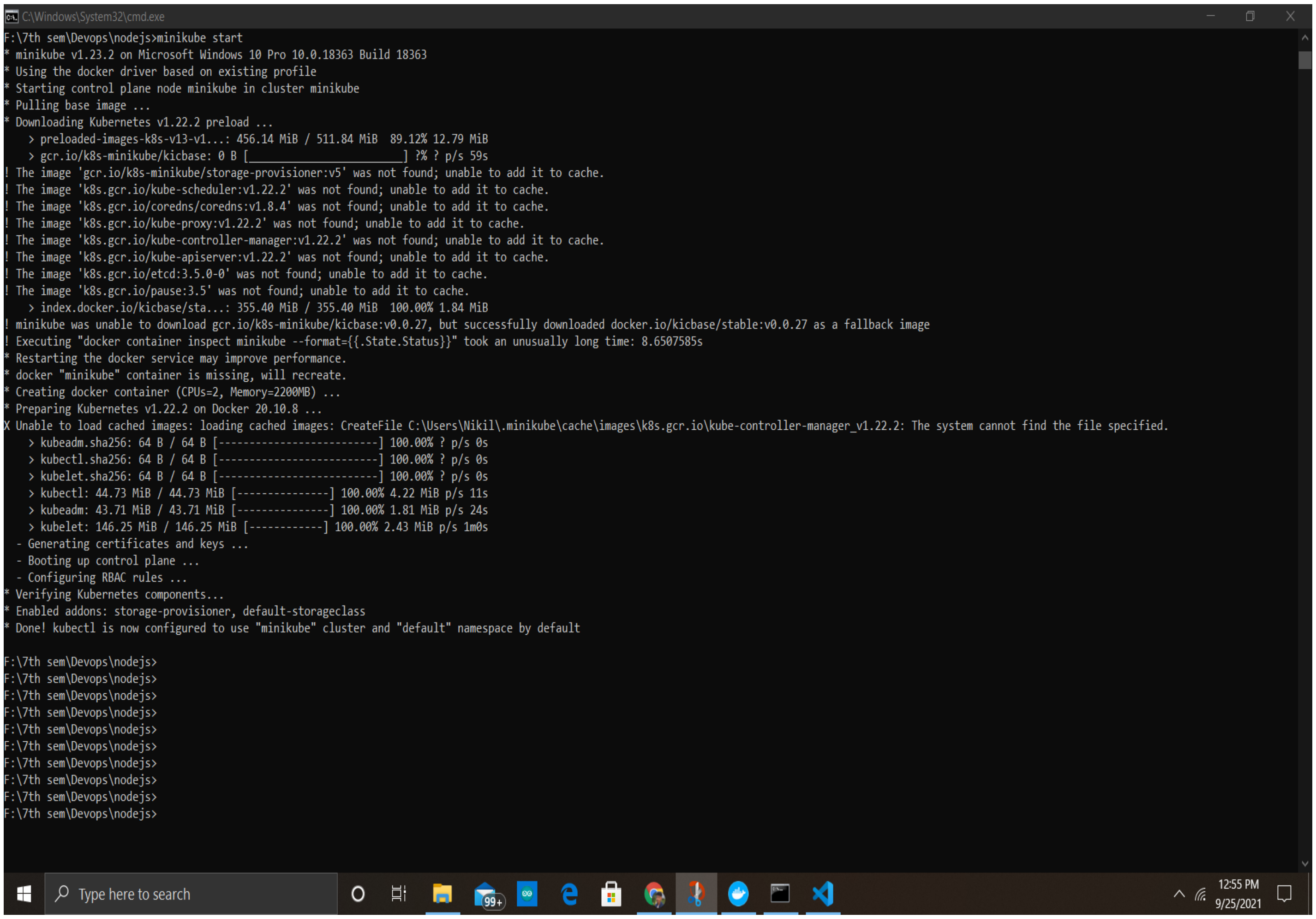
TAG	OS	PULLED	PUSHED
🟩 latest		15 minutes ago	15 minutes ago
🟩 1.1		15 minutes ago	a few second...

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

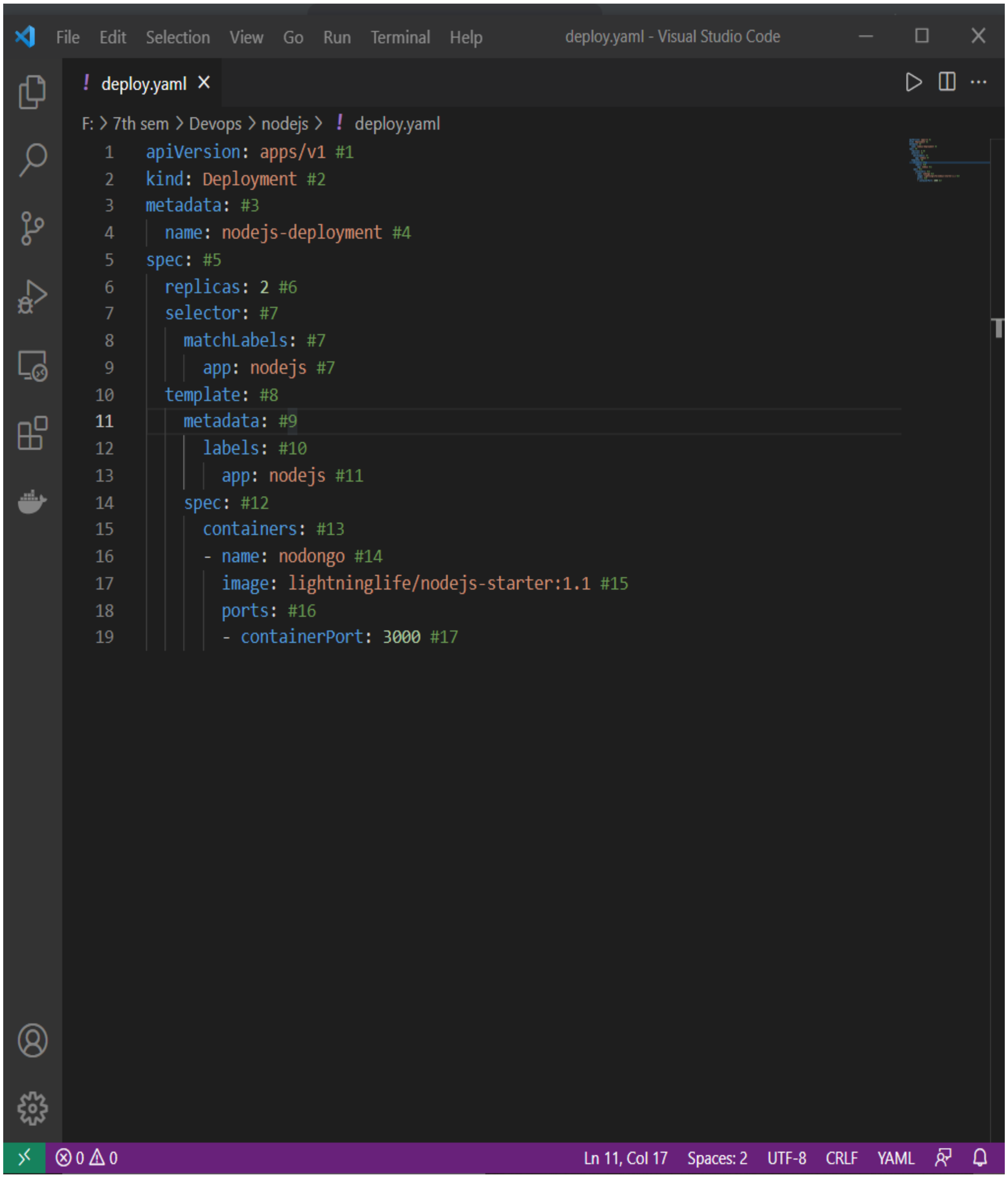
Available on Pro and Team plans.

Step10: Start the Kubernetes Cluster.



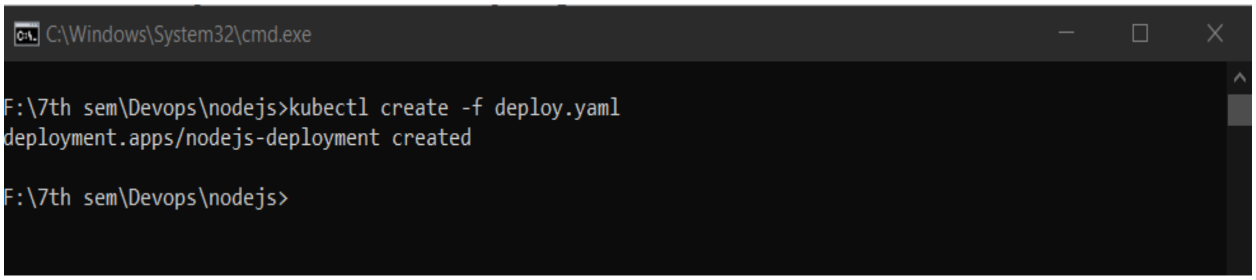
```
F:\7th sem\Devops\nodejs>minikube start
* minikube v1.23.2 on Microsoft Windows 10 Pro 10.0.18363 Build 18363
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.22.2 preload ...
    > preloaded-images-k8s-v13-v1...: 456.14 MiB / 511.84 MiB  89.12% 12.79 MiB
    > gcr.io/k8s-minikube/kicbase: 0 B [_______________________] ?% ? p/s 59s
! The image 'gcr.io/k8s-minikube/storage-provisioner:v5' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/kube-scheduler:v1.22.2' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/coredns/coredns:v1.8.4' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/kube-proxy:v1.22.2' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/kube-controller-manager:v1.22.2' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/kube-apiserver:v1.22.2' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/etcd:3.5.0-0' was not found; unable to add it to cache.
! The image 'k8s.gcr.io/pause:3.5' was not found; unable to add it to cache.
    > index.docker.io/kicbase/sta...: 355.40 MiB / 355.40 MiB  100.00% 1.84 MiB
! minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.27, but successfully downloaded docker.io/kicbase/stable:v0.0.27 as a fallback image
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 8.6507585s
* Restarting the docker service may improve performance.
* docker "minikube" container is missing, will recreate.
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.22.2 on Docker 20.10.8 ...
X Unable to load cached images: loading cached images: CreateFile C:\Users\Nikil\.minikube\cache\images\k8s.gcr.io\kube-controller-manager_v1.22.2: The system cannot find the file specified.
    > kubeadm.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubectl.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubelet.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubectl: 44.73 MiB / 44.73 MiB [---------------] 100.00% 4.22 MiB p/s 11s
    > kubeadm: 43.71 MiB / 43.71 MiB [---------------] 100.00% 1.81 MiB p/s 24s
    > kubelet: 146.25 MiB / 146.25 MiB [------------] 100.00% 2.43 MiB p/s 1m0s
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
F:\7th sem\Devops\nodejs>
```

Step11: Define YAML File To Create A Deployment In Kubernetes Cluster.


File Edit Selection View Go Run Terminal Help
deploy.yaml - Visual Studio Code
deploy.yaml
F: > 7th sem > Devops > nodejs > deploy.yaml
Ln 11, Col 17 Spaces: 2 UTF-8 CRLF YAML

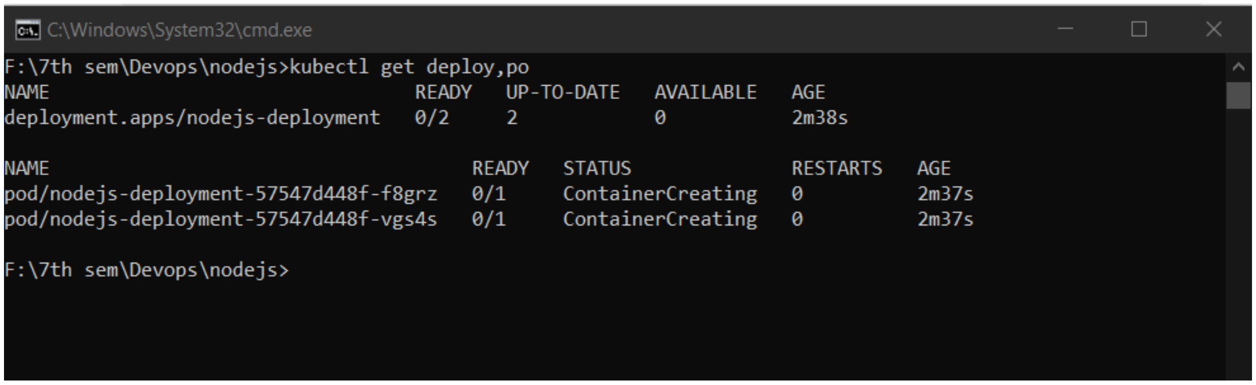
```yaml
apiVersion: apps/v1 #1
kind: Deployment #2
metadata: #3
  name: nodejs-deployment #4
spec: #5
  replicas: 2 #6
  selector: #7
    matchLabels: #7
      app: nodejs #7
  template: #8
    metadata: #9
      labels: #10
        app: nodejs #11
    spec: #12
      containers: #13
      - name: nodongo #14
        image: lightninglife/nodejs-starter:1.1 #15
        ports: #16
        - containerPort: 3000 #17
```

Step12: Create Deployment In Kubernetes Cluster.



C:\Windows\System32\cmd.exe

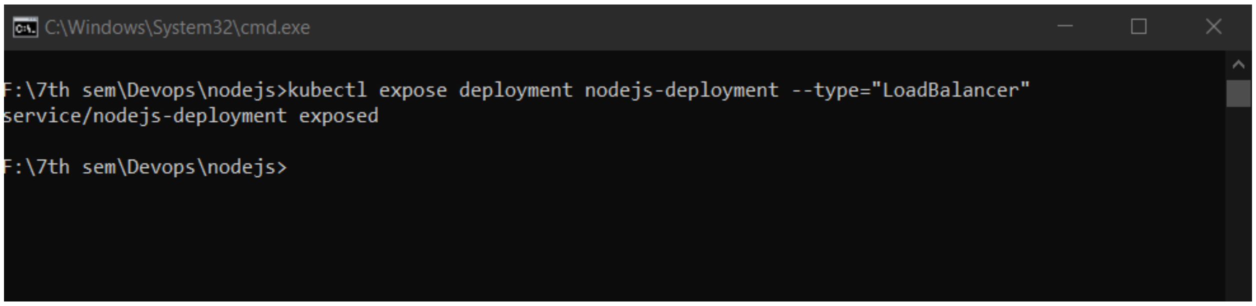
```
F:\7th sem\Devops\nodejs>kubectl create -f deploy.yaml
deployment.apps/nodejs-deployment created

F:\7th sem\Devops\nodejs>
```



C:\Windows\System32\cmd.exe

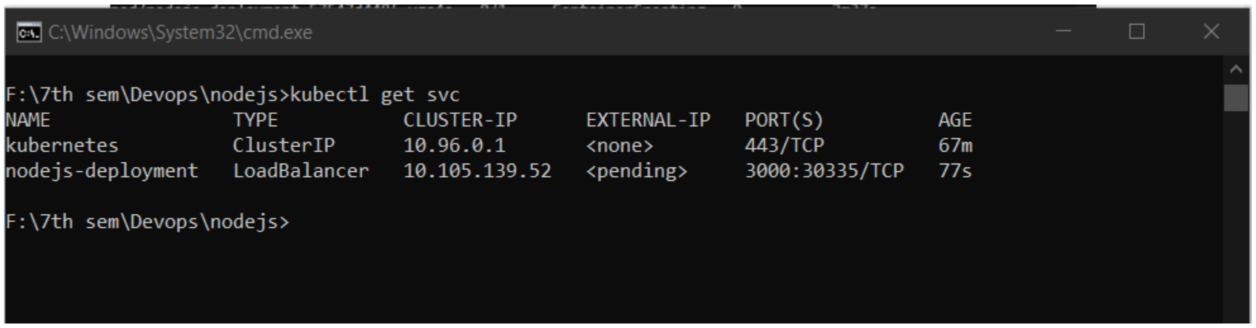
```
F:\7th sem\Devops\nodejs>kubectl get deploy,po
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-deployment   0/2     2            0           2m38s

NAME                                     READY   STATUS              RESTARTS   AGE
pod/nodejs-deployment-57547d448f-f8grz   0/1     ContainerCreating   0          2m37s
pod/nodejs-deployment-57547d448f-vgs4s   0/1     ContainerCreating   0          2m37s

F:\7th sem\Devops\nodejs>
```

Step13: Expose the deployment to the internet.



C:\Windows\System32\cmd.exe

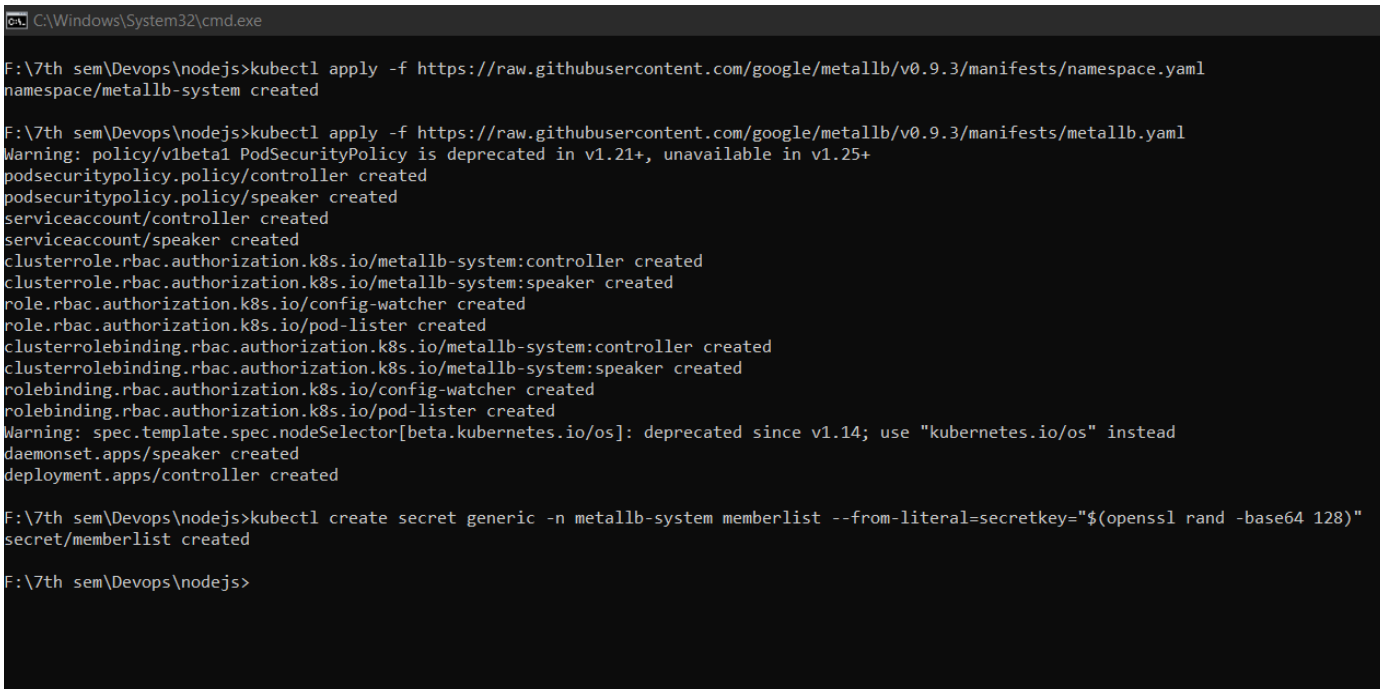
```
F:\7th sem\Devops\nodejs>kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed

F:\7th sem\Devops\nodejs>
```



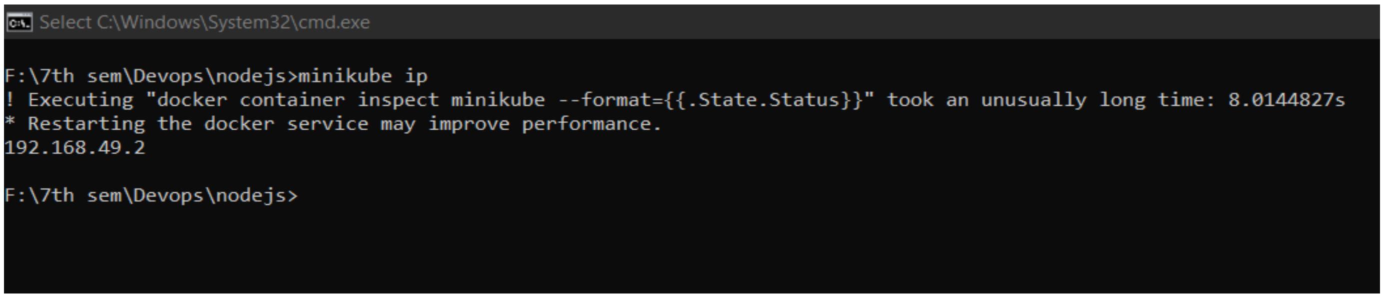
C:\Windows\System32\cmd.exe

```
F:\7th sem\Devops\nodejs>kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1       <none>        443/TCP          67m
nodejs-deployment   LoadBalancer   10.105.139.52   <pending>     3000:30335/TCP   77s

F:\7th sem\Devops\nodejs>
```

Step14: Using MetalLB In Your Minikube Environment.

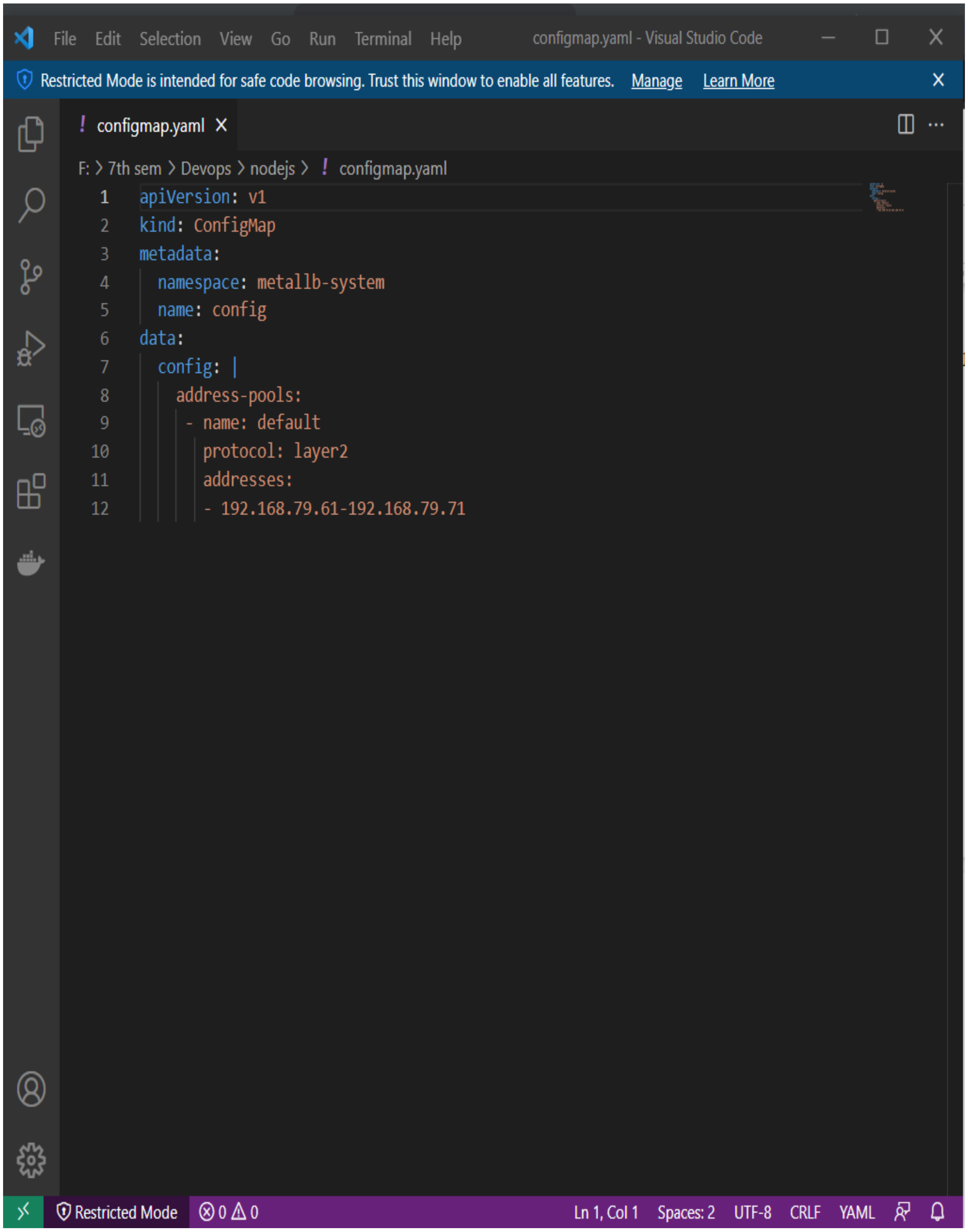


```
C:\Windows\System32\cmd.exe

F:\7th sem\Devops\nodejs>kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespace.yaml
namespace/metallb-system created

F:\7th sem\Devops\nodejs>kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/metallb.yaml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14; use "kubernetes.io/os" instead
daemonset.apps/speaker created
deployment.apps/controller created

F:\7th sem\Devops\nodejs>kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created

F:\7th sem\Devops\nodejs>
```



```
Select C:\Windows\System32\cmd.exe

F:\7th sem\Devops\nodejs>minikube ip
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 8.0144827s
* Restarting the docker service may improve performance.
192.168.49.2

F:\7th sem\Devops\nodejs>
```

After this, we'll create a config map for the address pool by running the following command:
code configmap.yaml

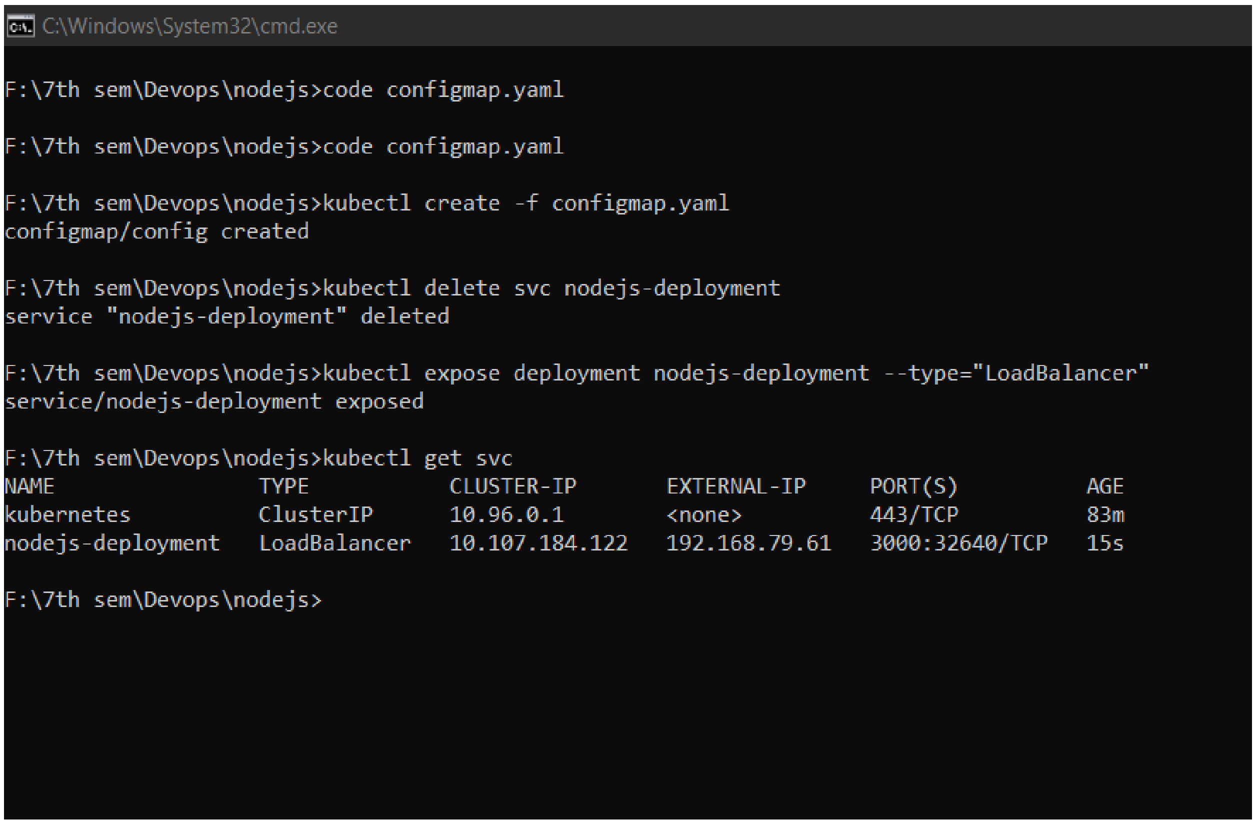

File Edit Selection View Go Run Terminal Help
configmap.yaml - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
configmap.yaml
F: > 7th sem > Devops > nodejs > configmap.yaml
Restricted Mode
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF YAML

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 192.168.79.61-192.168.79.71
```

In this configuration, MetalLB is instructed to hand out addresses from 192.168.79.61 to 192.168.79.71. After that, we’ll create a config map in the metallb-system namespace.

```
kubectl create -f configmap.yaml
kubectl delete svc nodejs-deployment
kubectl expose deployment nodejs-deployment--type="LoadBalancer"
```

Now that’s done, you’ll be getting External IP.



C:\Windows\System32\cmd.exe

```
F:\7th sem\Devops\nodejs>code configmap.yaml

F:\7th sem\Devops\nodejs>code configmap.yaml

F:\7th sem\Devops\nodejs>kubectl create -f configmap.yaml
configmap/config created

F:\7th sem\Devops\nodejs>kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted

F:\7th sem\Devops\nodejs>kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed

F:\7th sem\Devops\nodejs>kubectl get svc
NAME                TYPE           CLUSTER-IP       EXTERNAL-IP     PORT(S)          AGE
kubernetes          ClusterIP      10.96.0.1        <none>          443/TCP          83m
nodejs-deployment   LoadBalancer   10.107.184.122   192.168.79.61   3000:32640/TCP   15s

F:\7th sem\Devops\nodejs>
```

Thank you!