

## Project Steps

### 1. Set Up Git Version Control

#### 1.1. Initialize a Git Repository

Create a new directory for your project:

```
mkdir nodejs-k8s-project
```

```
cd nodejs-k8s-project
```

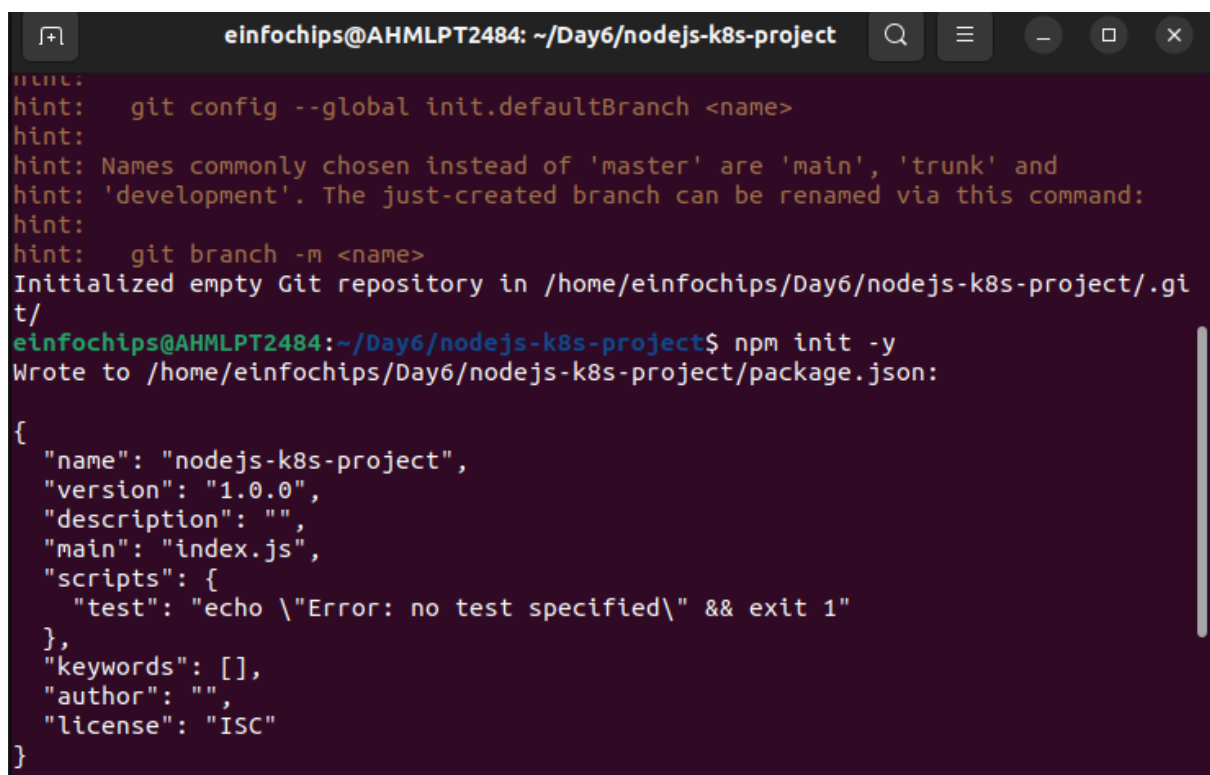
Initialize a Git repository:

```
git init
```

#### 1.2. Create a Node.js Application

Initialize a Node.js project:

```
npm init -y
```

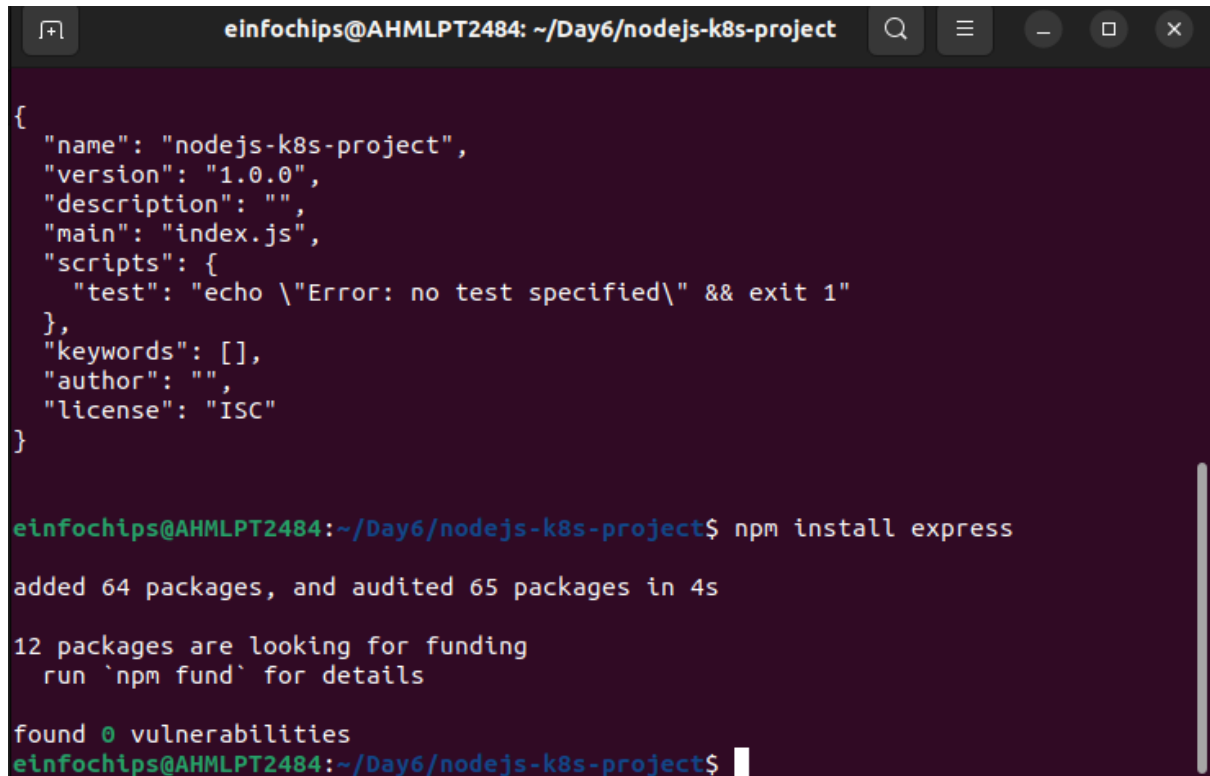
A terminal window with a dark background and light-colored text. The window title is 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project'. The terminal shows the output of 'git init', which includes several hints about branch naming and the creation of an empty Git repository. Below this, the command 'npm init -y' is executed, resulting in the creation of a 'package.json' file. The content of 'package.json' is displayed in a JSON format, showing fields like 'name', 'version', 'description', 'main', 'scripts', 'keywords', 'author', and 'license'.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/einfochips/Day6/nodejs-k8s-project/.git/
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ npm init -y
Wrote to /home/einfochips/Day6/nodejs-k8s-project/package.json:

{
  "name": "nodejs-k8s-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Install Express.js:

```
npm install express
```

A terminal window with a dark purple background. The title bar shows the user 'einfochips' on host 'AHMLPT2484' in the directory '~/Day6/nodejs-k8s-project'. The terminal displays the content of a package.json file, followed by the command 'npm install express' and its output. The output indicates that 64 packages were added and audited in 4 seconds, with 12 packages looking for funding and 0 vulnerabilities found.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
{
  "name": "nodejs-k8s-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ npm install express

added 64 packages, and audited 65 packages in 4s

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

found 0 vulnerabilities
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$
```

Create an `index.js` file with the following content:

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

const app = express();

const port = 3000;

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

  res.send('Hello, Kubernetes!');

});

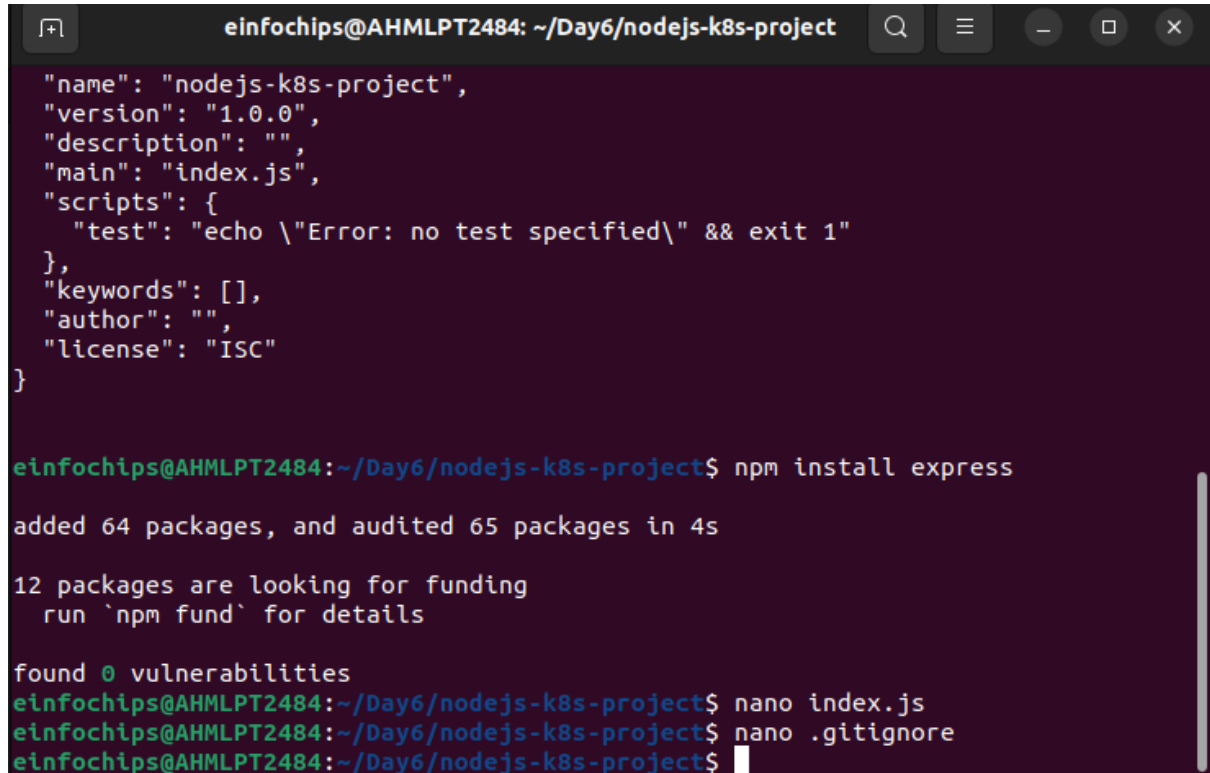
app.listen(port, () => {

  console.log(`App running at http://localhost:${port}`);
```

```
});
```

Create a `.gitignore` file to ignore `node_modules`:

```
node_modules
```

A terminal window with a dark purple background. The title bar shows 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project'. The terminal content shows a JSON object for package.json, followed by the command 'npm install express' and its output. Then, the command 'nano index.js' is entered, followed by 'nano .gitignore', and the prompt returns.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
{
  "name": "nodejs-k8s-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ npm install express
added 64 packages, and audited 65 packages in 4s

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

found 0 vulnerabilities
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano .gitignore
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$
```

### 1.3. Commit the Initial Code

Add files to Git:

```
git add .
```

Commit the changes:

```
git commit -m "Initial commit with Node.js app"
```

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
"author": "",
"license": "ISC"
}

einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ npm install express

added 64 packages, and audited 65 packages in 4s

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

found 0 vulnerabilities
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano .gitignore
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git add .
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git commit -m "Initial commit w
ith Node.js app"
[master (root-commit) f0ea9ae] Initial commit with Node.js app
4 files changed, 1213 insertions(+)
create mode 100644 .gitignore
create mode 100644 index.js
create mode 100644 package-lock.json
create mode 100644 package.json
```

## 2. Branching and Fast-Forward Merge

### 2.1. Create a New Branch

Create and switch to a new branch `feature/add-route`:

```
git checkout -b feature/add-route
```

### 2.2. Implement a New Route

Modify `index.js` to add a new route:

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

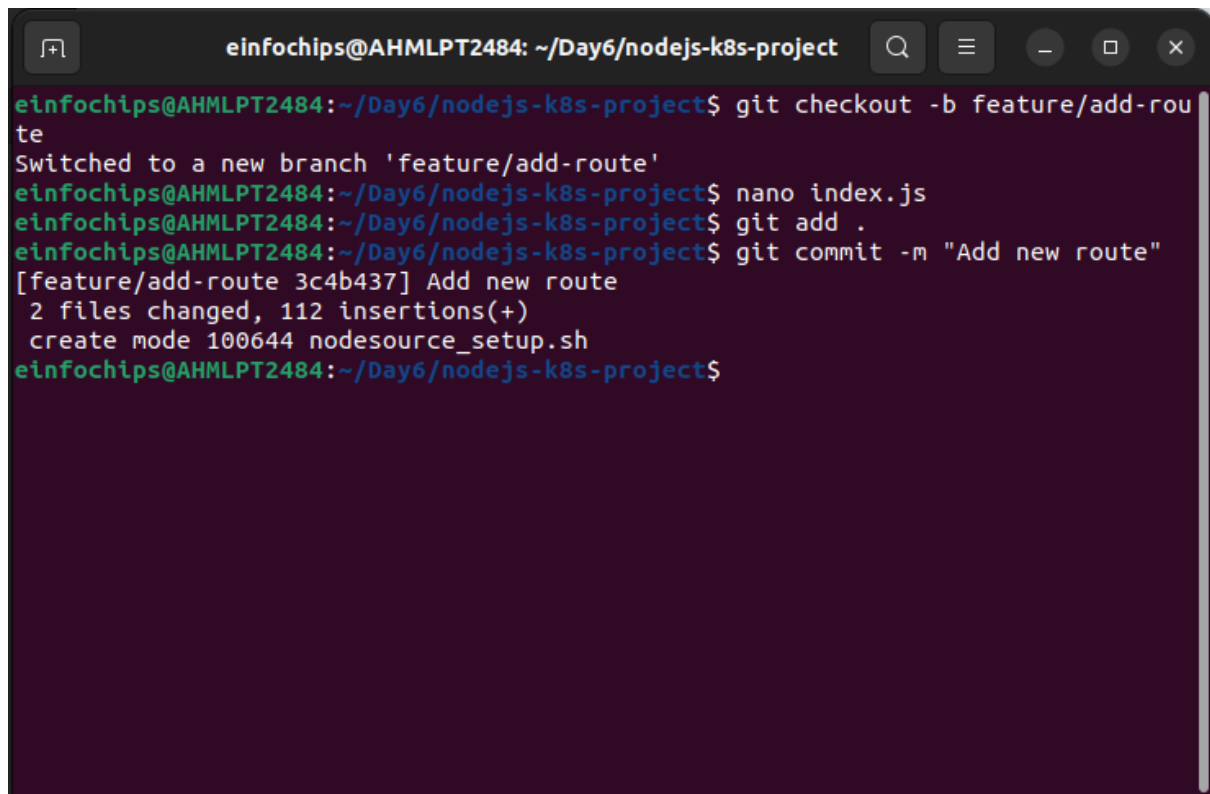
    res.send('This is a new route!');

});
```

Commit the changes:

```
git add .
```

```
git commit -m "Add new route"
```

A terminal window with a dark background and light green text. The window title is 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project'. The terminal shows the following commands and output: 

```
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git checkout -b feature/add-route
Switched to a new branch 'feature/add-route'
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git add .
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git commit -m "Add new route"
[feature/add-route 3c4b437] Add new route
 2 files changed, 112 insertions(+)
 create mode 100644 nodesource_setup.sh
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$
```

### 2.3. Merge the Branch Using Fast-Forward

Switch back to the `main` branch:

```
git checkout main
```

Merge the `feature/add-route` branch using fast-forward:

```
git merge --ff-only feature/add-route
```

Delete the feature branch:

```
git branch -d feature/add-route
```

## 3. Containerize the Node.js Application

### 3.1. Create a Dockerfile

Create a `Dockerfile` with the following content:

```
FROM node:14
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY . .
```

```
EXPOSE 3000
```

```
CMD ["node", "index.js"]
```

### 3.2. Build and Test the Docker Image

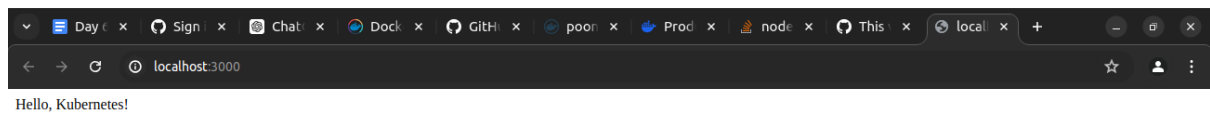
Build the Docker image:

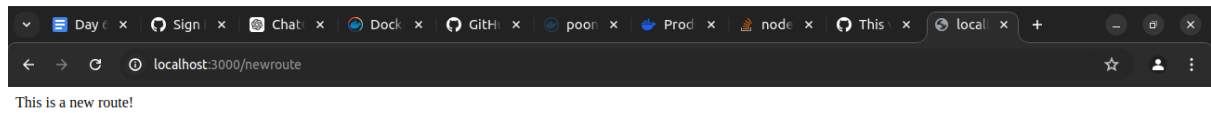
```
docker build -t nodejs-k8s-app .
```

Run the Docker container to test:

```
docker run -p 3000:3000 nodejs-k8s-app
```

1. Access <http://localhost:3000> to see the app running.





## 4. Deploying to Minikube Kubernetes

### 4.1. Start Minikube

Start Minikube:

```
minikube start
```

### 4.2. Create Kubernetes Deployment and Service Manifests

Create a `deployment.yaml` file:

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: nodejs-app
```

```
spec:
```

```
  replicas: 2
```

```
selector:

  matchLabels:

    app: nodejs-app

template:

  metadata:

    labels:

      app: nodejs-app

  spec:

    containers:

      - name: nodejs-app

        image: nodejs-k8s-app:latest

        ports:

          - containerPort: 3000
```

Create a `service.yaml` file for ClusterIP:

```
apiVersion: v1

kind: Service

metadata:

  name: nodejs-service

spec:
```



```
selector:
```

```
  app: nodejs-app
```

```
ports:
```

```
- protocol: TCP
```

```
  port: 80
```

```
  targetPort: 3000
```

```
type: ClusterIP
```

Create a `service-nodeport.yaml` file for NodePort:

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: nodejs-service-nodeport
```

```
spec:
```

```
  selector:
```

```
    app: nodejs-app
```

```
  ports:
```

```
- protocol: TCP
```

```
  port: 80
```

```
  targetPort: 3000
```

```
nodePort: 30001
```

```
type: NodePort
```

#### 4.3. Apply Manifests to Minikube

Apply the deployment:

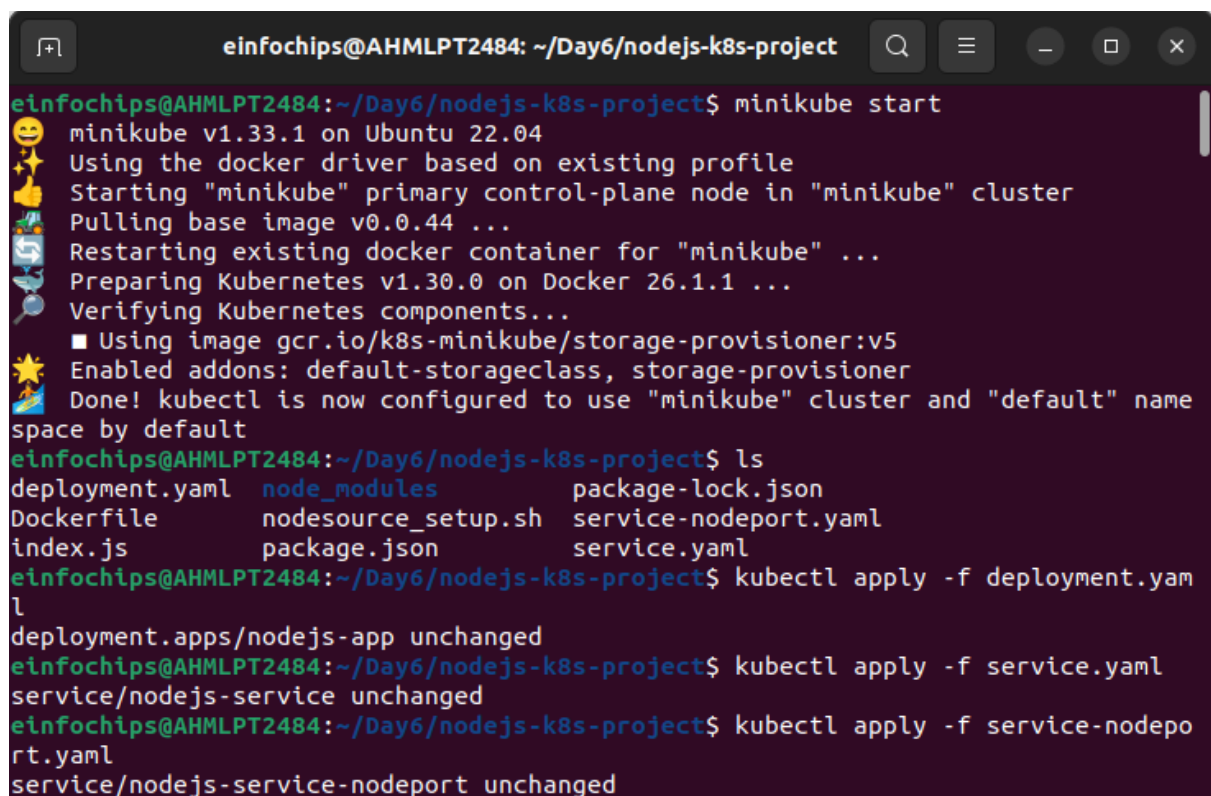
```
kubectl apply -f deployment.yaml
```

Apply the ClusterIP service:

```
kubectl apply -f service.yaml
```

Apply the NodePort service:

```
kubectl apply -f service-nodeport.yaml
```

A terminal window titled 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project' showing the execution of minikube and kubectl commands. The minikube start command outputs various status messages including version, driver, and component verification. The kubectl apply commands show that the deployment, service, and service-nodeport manifests are unchanged.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ minikube start
🐹 minikube v1.33.1 on Ubuntu 22.04
🔧 Using the docker driver based on existing profile
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.44 ...
🔄 Restarting existing docker container for "minikube" ...
🔧 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
🔍 Verifying Kubernetes components...
   ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: default-storageclass, storage-provisioner
🏁 Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ ls
deployment.yaml  node_modules  package-lock.json
Dockerfile      nodesource_setup.sh  service-nodeport.yaml
index.js        package.json  service.yaml
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl apply -f deployment.yam
l
deployment.apps/nodejs-app unchanged
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl apply -f service.yaml
service/nodejs-service unchanged
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl apply -f service-nodepo
rt.yaml
service/nodejs-service-nodeport unchanged
```

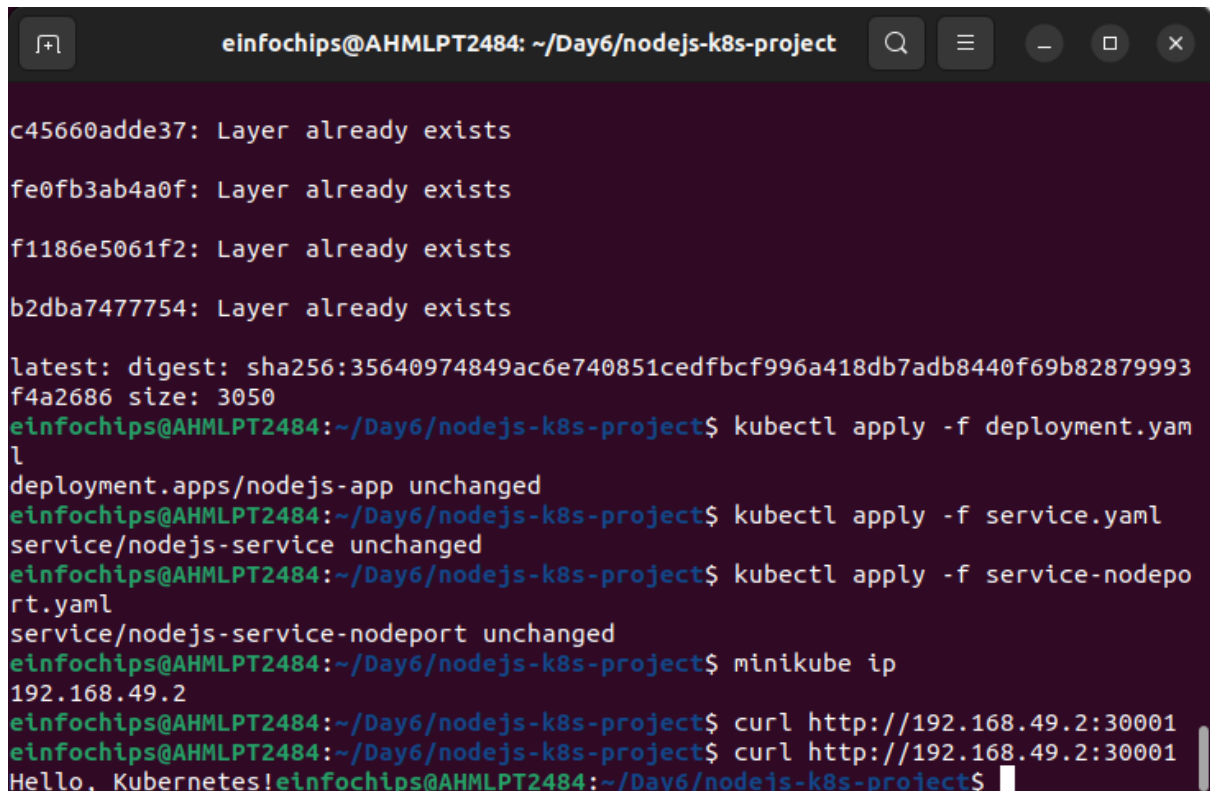
#### 4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

1. Access the application using the NodePort:

```
curl http://<minikube-ip>:30001
```

A terminal window titled 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project' with standard window controls. The terminal shows the following commands and output:  
c45660adde37: Layer already exists  
fe0fb3ab4a0f: Layer already exists  
f1186e5061f2: Layer already exists  
b2dba7477754: Layer already exists  
latest: digest: sha256:35640974849ac6e740851cedfbcf996a418db7adb8440f69b82879993f4a2686 size: 3050  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ kubectl apply -f deployment.yaml  
deployment.apps/nodejs-app unchanged  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ kubectl apply -f service.yaml  
service/nodejs-service unchanged  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ kubectl apply -f service-nodeport.yaml  
service/nodejs-service-nodeport unchanged  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ minikube ip  
192.168.49.2  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ curl http://192.168.49.2:30001  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$ curl http://192.168.49.2:30001  
Hello, Kubernetes!  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project\$

## Making Changes to the App and Redeploying Using Kubernetes

### 6. Making Changes to the Node.js Application

#### 6.1. Create a New Branch for Changes

Create and switch to a new branch `feature/update-message`:

```
git checkout -b feature/update-message
```

#### 6.2. Update the Application

Modify `index.js` to change the message:

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

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

```
const port = 3000;
```

```

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

    res.send('Hello, Kubernetes! Updated version.');
```

});

```

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

    res.send('This is a new route!');
```

});

```

app.listen(port, () => {

    console.log(`App running at http://localhost:${port}`);

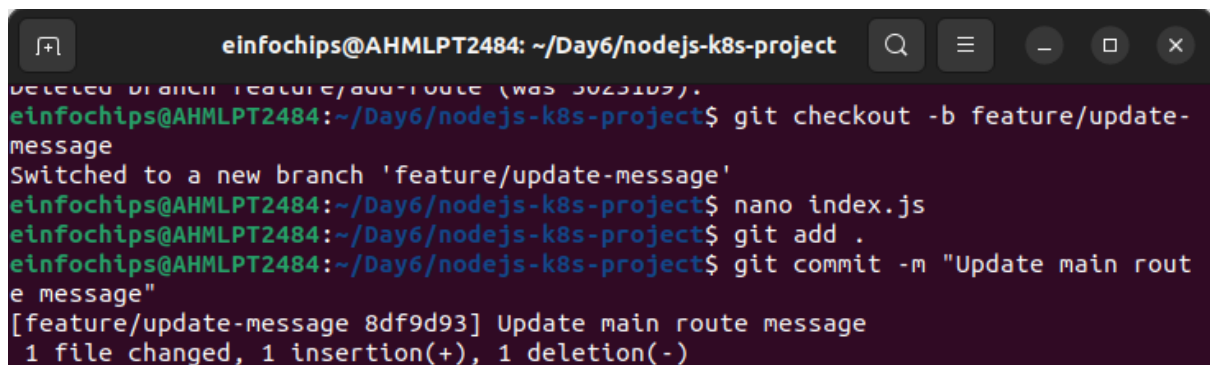
});
```

### 6.3. Commit the Changes

Add and commit the changes:

```
git add .
```

```
git commit -m "Update main route message"
```



```

einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
Deleted branch feature/add-route (was 3023109).
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git checkout -b feature/update-
message
Switched to a new branch 'feature/update-message'
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git add .
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ git commit -m "Update main rout
e message"
[feature/update-message 8df9d93] Update main route message
1 file changed, 1 insertion(+), 1 deletion(-)
```

## 7. Merge the Changes and Rebuild the Docker Image

### 7.1. Merge the Feature Branch

Switch back to the `main` branch:

```
git checkout main
```

Merge the `feature/update-message` branch:

```
git merge --ff-only feature/update-message
```

Delete the feature branch:

```
git branch -d feature/update-message
```

A terminal window with a dark background and light-colored text. The window title is 'einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project'. The terminal shows the following commands and output: 'git checkout master' results in 'Switched to branch 'master''; 'git merge --ff-only feature/update-message' results in 'Updating 50231b9..8df9d93' and 'Fast-forward'; a diff shows 'index.js | 2 +-'; '1 file changed, 1 insertion(+), 1 deletion(-)'; and 'git branch -d feature/update-message' results in 'Deleted branch feature/update-message (was 8df9d93)'.

## 7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

```
docker build -t nodejs-k8s-app:v2 .
```

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
[...]  
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ docker build -t nodejs-k8s-app:v2 .  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
Install the buildx component to build images with BuildKit:  
https://docs.docker.com/go/buildx/  
Sending build context to Docker daemon 2.762MB  
Step 1/7 : FROM node:14  
--> 1d12470fa662  
Step 2/7 : WORKDIR /app  
--> Using cache  
--> 117ee17d1eeb  
Step 3/7 : COPY package*.json ./  
--> Using cache  
--> 4d50f404c33e  
Step 4/7 : RUN npm install  
--> Using cache  
--> 01c298e6818a  
Step 5/7 : COPY . .  
--> 91ad5a50ae40  
Step 6/7 : EXPOSE 3000  
--> Running in 9bd3db1d27a1  
Removing intermediate container 9bd3db1d27a1
```

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project  
Sending build context to Docker daemon 2.762MB  
Step 1/7 : FROM node:14  
--> 1d12470fa662  
Step 2/7 : WORKDIR /app  
--> Using cache  
--> 117ee17d1eeb  
Step 3/7 : COPY package*.json ./  
--> Using cache  
--> 4d50f404c33e  
Step 4/7 : RUN npm install  
--> Using cache  
--> 01c298e6818a  
Step 5/7 : COPY . .  
--> 91ad5a50ae40  
Step 6/7 : EXPOSE 3000  
--> Running in 9bd3db1d27a1  
Removing intermediate container 9bd3db1d27a1  
--> d5f16f6c3620  
Step 7/7 : CMD ["node", "index.js"]  
--> Running in 11d8030f8940  
Removing intermediate container 11d8030f8940  
--> ea3c606d83b4  
Successfully built ea3c606d83b4  
Successfully tagged nodejs-k8s-app:v2
```

## 8. Update Kubernetes Deployment

### 8.1. Update the Deployment Manifest

Modify `deployment.yaml` to use the new image version:

```
apiVersion: apps/v1

kind: Deployment

metadata:

  name: nodejs-app

spec:

  replicas: 2

  selector:

    matchLabels:

      app: nodejs-app

  template:

    metadata:

      labels:

        app: nodejs-app

    spec:

      containers:

        - name: nodejs-app

          image: nodejs-k8s-app:v2
```

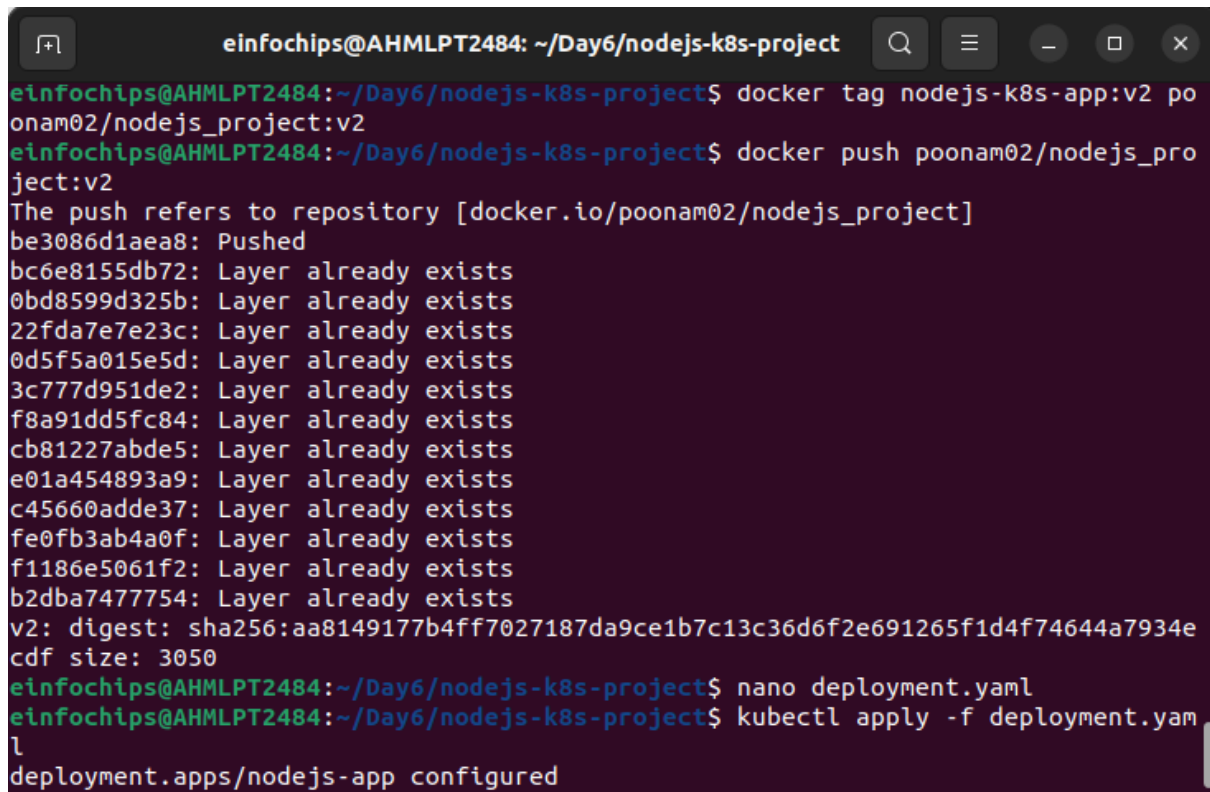
ports:

- containerPort: 3000

## 8.2. Apply the Updated Manifest

Apply the updated deployment:

```
kubectl apply -f deployment.yaml
```



```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ docker tag nodejs-k8s-app:v2 poonam02/nodejs_project:v2
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ docker push poonam02/nodejs_project:v2
The push refers to repository [docker.io/poonam02/nodejs_project]
be3086d1aea8: Pushed
bc6e8155db72: Layer already exists
0bd8599d325b: Layer already exists
22fda7e7e23c: Layer already exists
0d5f5a015e5d: Layer already exists
3c777d951de2: Layer already exists
f8a91dd5fc84: Layer already exists
cb81227abde5: Layer already exists
e01a454893a9: Layer already exists
c45660adde37: Layer already exists
fe0fb3ab4a0f: Layer already exists
f1186e5061f2: Layer already exists
b2dba7477754: Layer already exists
v2: digest: sha256:aa8149177b4ff7027187da9ce1b7c13c36d6f2e691265f1d4f74644a7934ecdf size: 3050
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano deployment.yaml
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app configured
```

## 8.3. Verify the Update

Check the status of the deployment:

```
kubectl rollout status deployment/nodejs-app
```



```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
22fda7e7e23c: Layer already exists
0d5f5a015e5d: Layer already exists
3c777d951de2: Layer already exists
f8a91dd5fc84: Layer already exists
cb81227abde5: Layer already exists
e01a454893a9: Layer already exists
c45660adde37: Layer already exists
fe0fb3ab4a0f: Layer already exists
f1186e5061f2: Layer already exists
b2dba7477754: Layer already exists
v2: digest: sha256:aa8149177b4ff7027187da9ce1b7c13c36d6f2e691265f1d4f74644a7934e
cdf size: 3050
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ nano deployment.yaml
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app configured
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl rollout status deployment/nodejs-app
Waiting for deployment "nodejs-app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nodejs-app" rollout to finish: 1 old replicas are pending termination...
deployment "nodejs-app" successfully rolled out
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$
```

## 9. Access the Updated Application

### 9.1. Access Through ClusterIP Service

Forward the port to access the ClusterIP service:

```
kubectl port-forward service/nodejs-service 8080:80
```

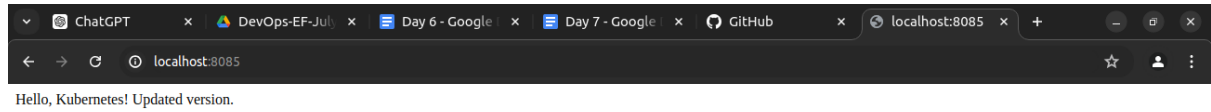
1. Open your browser and navigate to <http://localhost:8080> to see the updated message.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
deployment nodejs-app successfully rolled out
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl port-forward service/nodejs-service 8080:80
Unable to listen on port 8080: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:8080: bind: address already in use unable to create listener: Error listen tcp6 [::1]:8080: bind: address already in use]
error: unable to listen on any of the requested ports: [{8080 3000}]
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
backend-5cf7cf7d5c-d7z92            0/1     ImagePullBackOff    0           35h
backend-5cf7cf7d5c-dll88            0/1     ImagePullBackOff    0           35h
db-99c49d8c6-tkpvs                 1/1     Running             4 (12h ago) 35h
frontend-76dc6978c-8txs5           1/1     Running             4 (12h ago) 35h
frontend-76dc6978c-szbgw           1/1     Running             4 (12h ago) 35h
nodejs-app-7f567b54c6-hq6h5        1/1     Running             0           14m
nodejs-app-7f567b54c6-zf6jf        1/1     Running             0           14m
webapp-ff7d56d67-m2kw7             1/1     Running             7 (12h ago) 2d
webapp-ff7d56d67-nfg5d             1/1     Running             7 (12h ago) 2d
webapp-ff7d56d67-vzfgx             1/1     Running             7 (12h ago) 2d
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl get services
NAME                                TYPE               CLUSTER-IP      EXTERNAL-IP      PORT(S)
kubernetes                         ClusterIP          10.96.0.1       <none>           443/TCP
```

Didn't run on 8080 so changed the port to 8085.

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
Type:                               ClusterIP
IP Family Policy:                   SingleStack
IP Families:                        IPv4
IP:                                 10.110.47.241
IPs:                                10.110.47.241
Port:                               <unset> 80/TCP
TargetPort:                         3000/TCP
Endpoints:                          10.244.0.71:3000,10.244.0.72:3000
Session Affinity:                   None
Events:                             <none>
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl port-forward service/nodejs-service 8081:80
Unable to listen on port 8081: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:8081: bind: address already in use unable to create listener: Error listen tcp6 [::1]:8081: bind: address already in use]
error: unable to listen on any of the requested ports: [{8081 3000}]
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ kubectl port-forward service/nodejs-service 8085:80
Forwarding from 127.0.0.1:8085 -> 3000
Forwarding from [::1]:8085 -> 3000
Handling connection for 8085
Handling connection for 8085
```

Output:



## 9.2. Access Through NodePort Service

1. Access the application using the NodePort:

```
curl http://<minikube-ip>:30001
```

```
einfochips@AHMLPT2484: ~/Day6/nodejs-k8s-project
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ minikube ip
192.168.49.2
einfochips@AHMLPT2484:~/Day6/nodejs-k8s-project$ curl http://192.168.49.2:30001
Hello, Kubernetes! Updated version.einfochips@AHMLPT2484:~/Day6/nodejs-k8s-proje
ct$
```