Project 01

Deploying a Node.js App Using Minikube Kubernetes

Overview

This project guides you through deploying a Node.js application using Minikube Kubernetes. You'll use Git for version control, explore branching and fast-forward merges, and set up Kubernetes services and deployment pods, including ClusterIP and NodePort service types.

Prerequisites

- Minikube installed
- kubectl installed
- Git installed
- Node.js installed (https://nodejs.org/en/download/package-manager/all#debian-and-ubuntu-based-linux-distributions)

Project Steps

1. Set Up Git Version Control

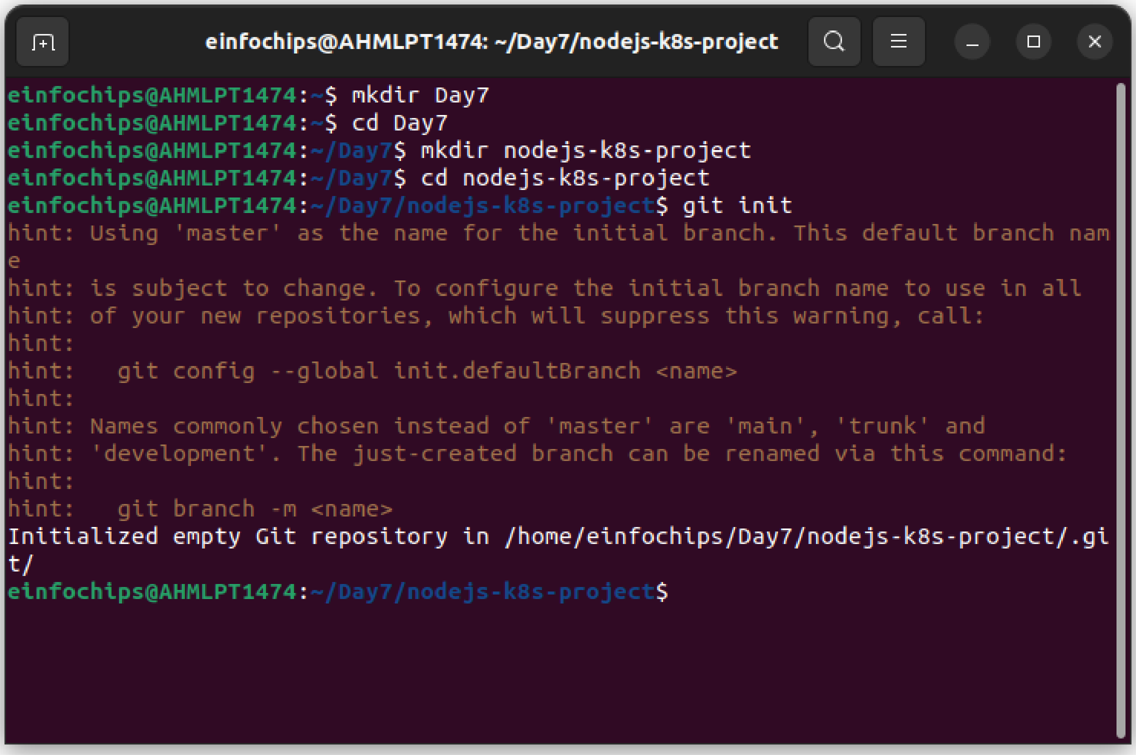
1.1. Initialize a Git Repository

Create a new directory for your project:

```
mkdir nodejs-k8s-project

cd nodejs-k8s-project
```

Initialize a Git repository:

```
git init
```



```
einfochips@AHMLPT1474:~$ mkdir Day7
einfochips@AHMLPT1474:~$ cd Day7
einfochips@AHMLPT1474:~/Day7$ mkdir nodejs-k8s-project
einfochips@AHMLPT1474:~/Day7$ cd nodejs-k8s-project
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git init
hint: Using 'master' as the name for the initial branch. This default branch nam
e
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
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/Day7/nodejs-k8s-project/.gi
t/
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$
```

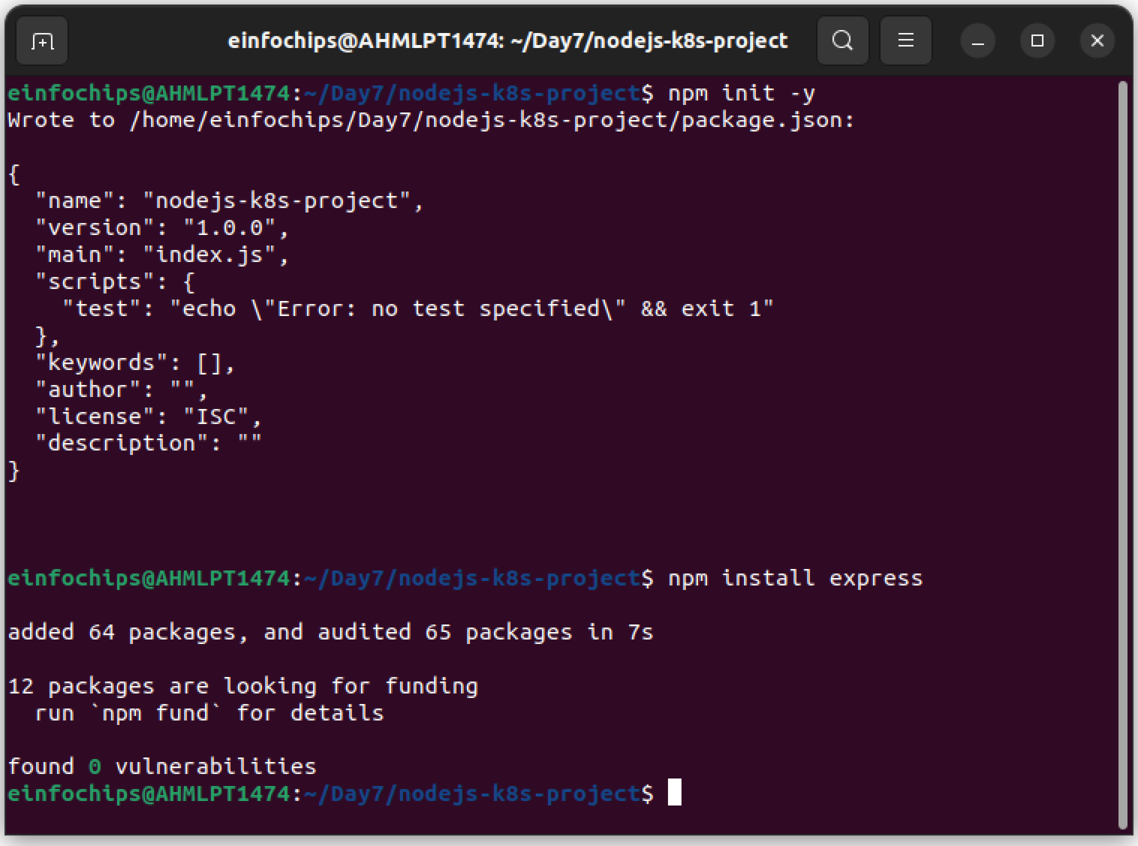
1.2. Create a Node.js Application

Initialize a Node.js project:

```
npm init -y
```

Install Express.js:

```
npm install express
```



einfochips@AHMLPT1474: ~/Day7/nodejs-k8s-project

```
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ npm init -y
Wrote to /home/einfochips/Day7/nodejs-k8s-project/package.json:

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



einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ npm install express

added 64 packages, and audited 65 packages in 7s

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

found 0 vulnerabilities
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ 
```

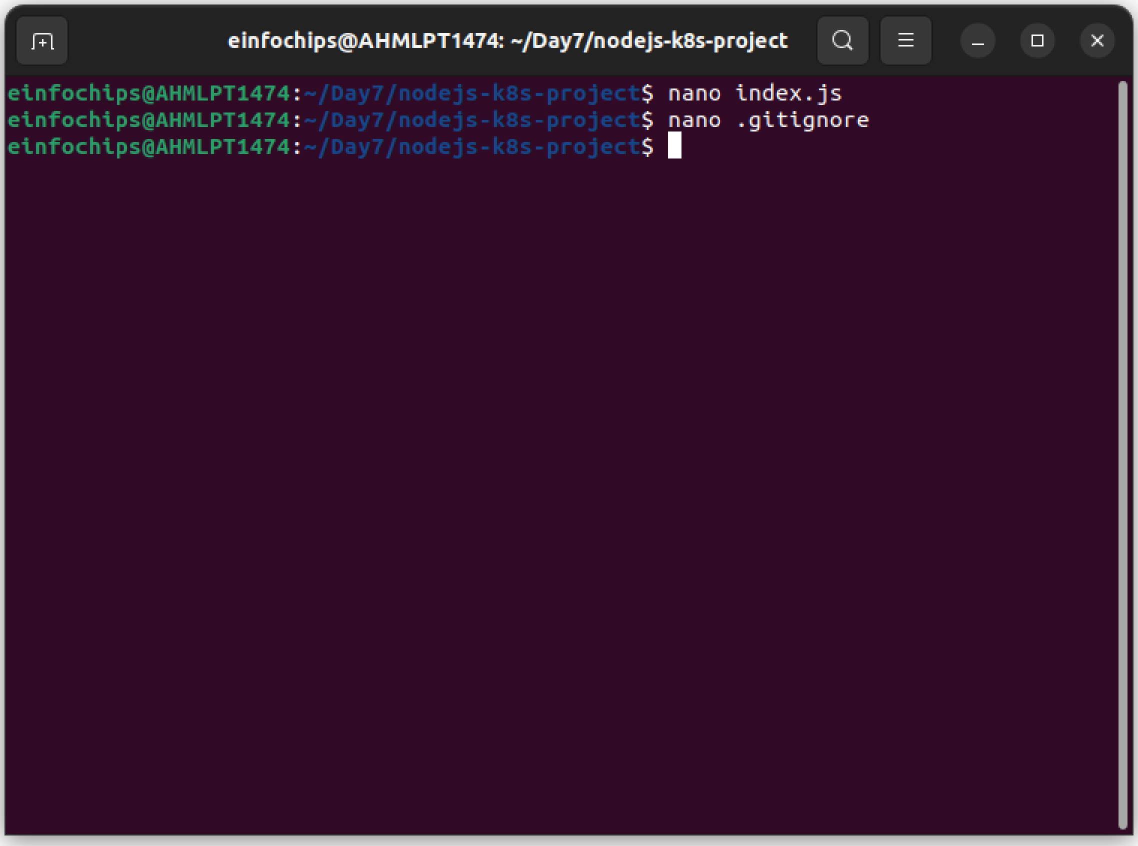
Create an index.js file with the following content:

```
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}`);

});
```

1.

Create a .gitignore file to ignore node_modules:

node_modules



```
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ nano .gitignore
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ 
```

1.3. Commit the Initial Code

Add files to Git:

git add .

Commit the changes:

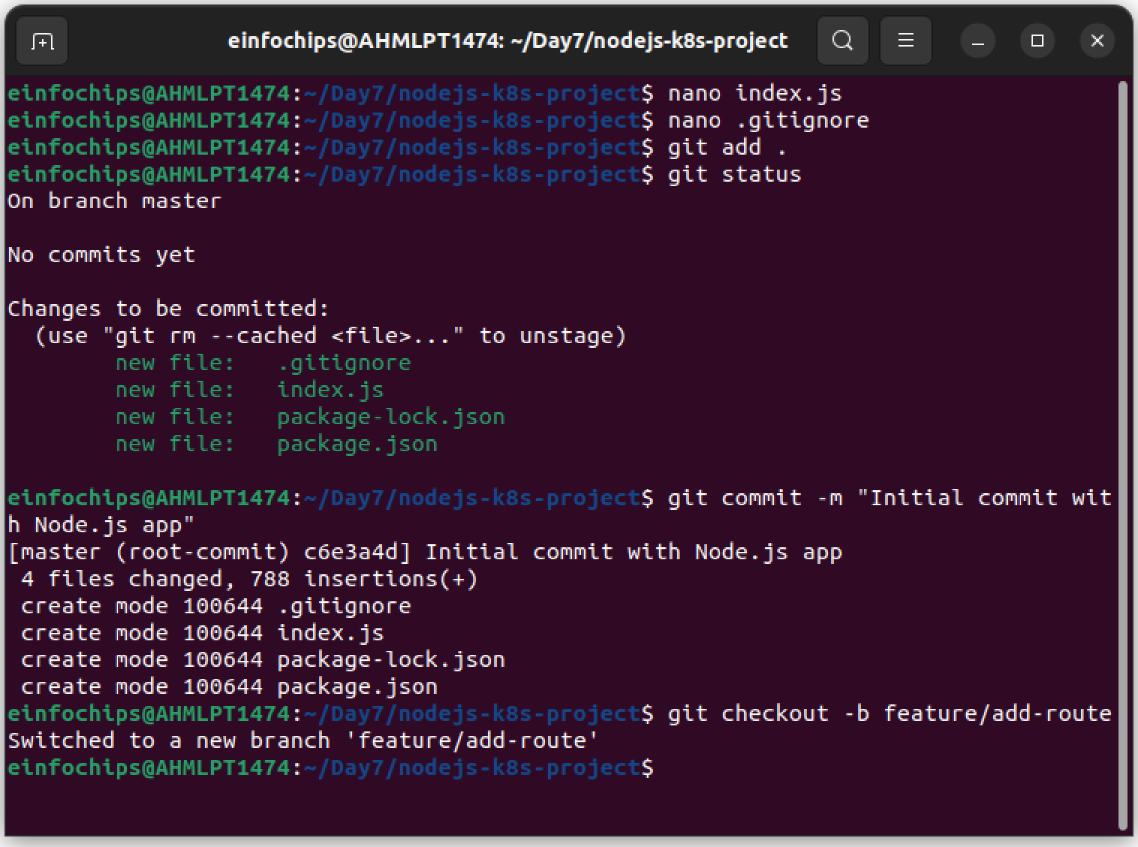
git commit -m "Initial commit with Node.js app"

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



```
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ nano .gitignore
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   index.js
        new file:   package-lock.json
        new file:   package.json

einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git commit -m "Initial commit wit
h Node.js app"
[master (root-commit) c6e3a4d] Initial commit with Node.js app
 4 files changed, 788 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 index.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git checkout -b feature/add-route
Switched to a new branch 'feature/add-route'
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$
```

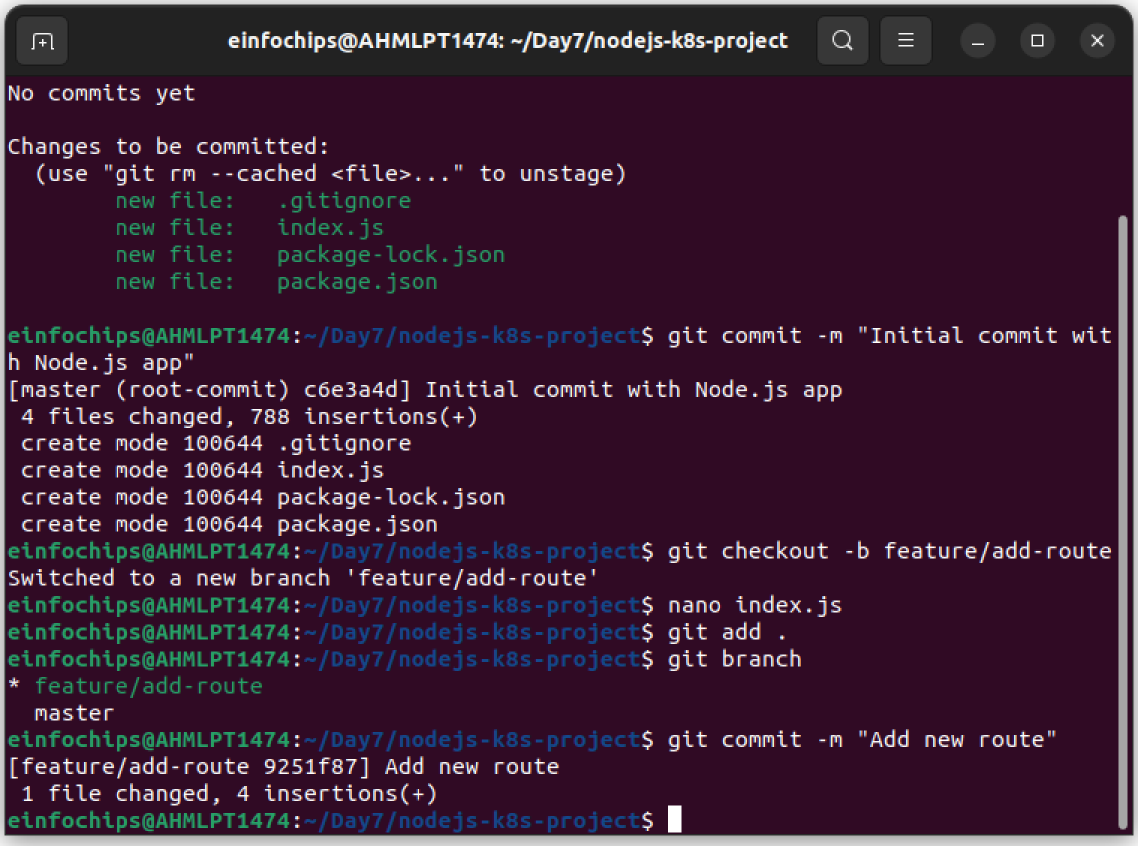
2.2. Implement a New Route

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"
```



```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   index.js
        new file:   package-lock.json
        new file:   package.json

einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git commit -m "Initial commit wit
h Node.js app"
[master (root-commit) c6e3a4d] Initial commit with Node.js app
 4 files changed, 788 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 index.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git checkout -b feature/add-route
Switched to a new branch 'feature/add-route'
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ nano index.js
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git branch
* feature/add-route
  master
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git commit -m "Add new route"
[feature/add-route 9251f87] Add new route
 1 file changed, 4 insertions(+)
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ 
```

2.3. Merge the Branch Using Fast-Forward

Switch back to the main branch:

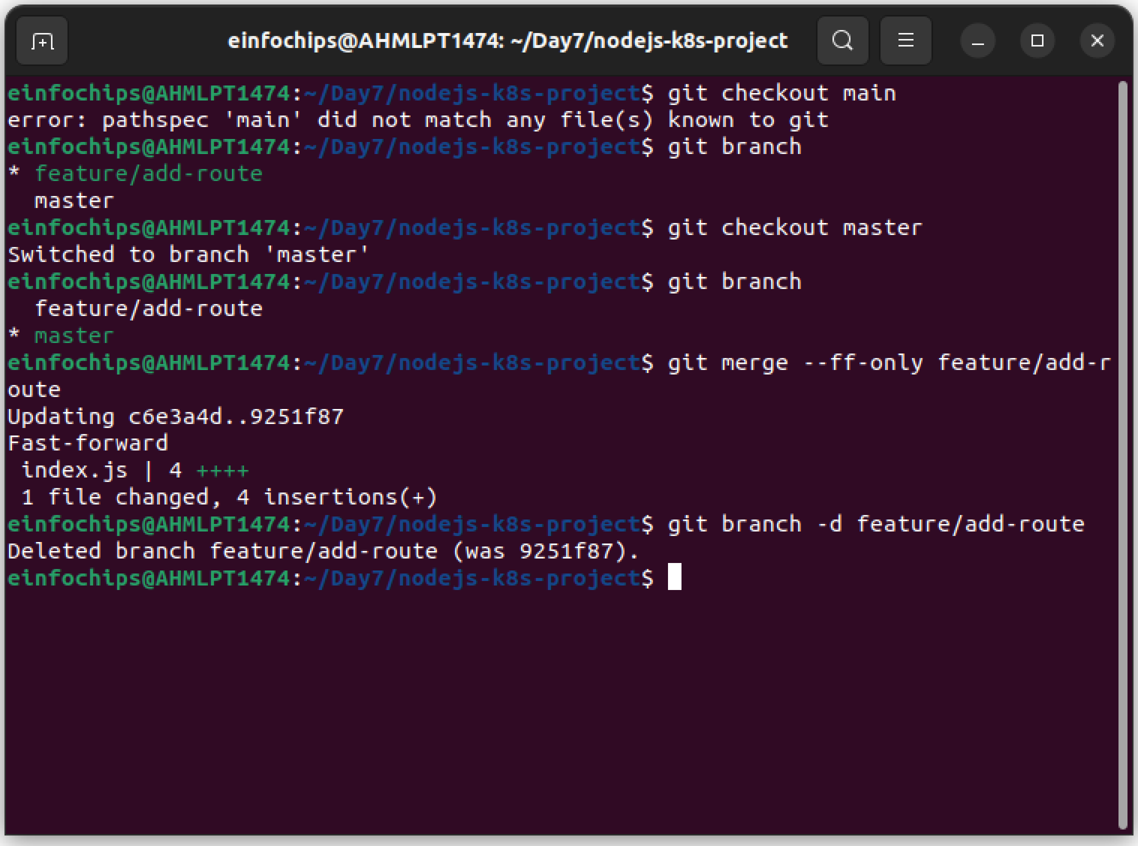
```
git checkout main
```

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
```



```
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git branch
* feature/add-route
  master
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git checkout master
Switched to branch 'master'
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git branch
  feature/add-route
* master
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git merge --ff-only feature/add-r
oute
Updating c6e3a4d..9251f87
Fast-forward
 index.js | 4 ++++
 1 file changed, 4 insertions(+)
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ git branch -d feature/add-route
Deleted branch feature/add-route (was 9251f87).
einfochips@AHMLPT1474:~/Day7/nodejs-k8s-project$ 
```

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
```

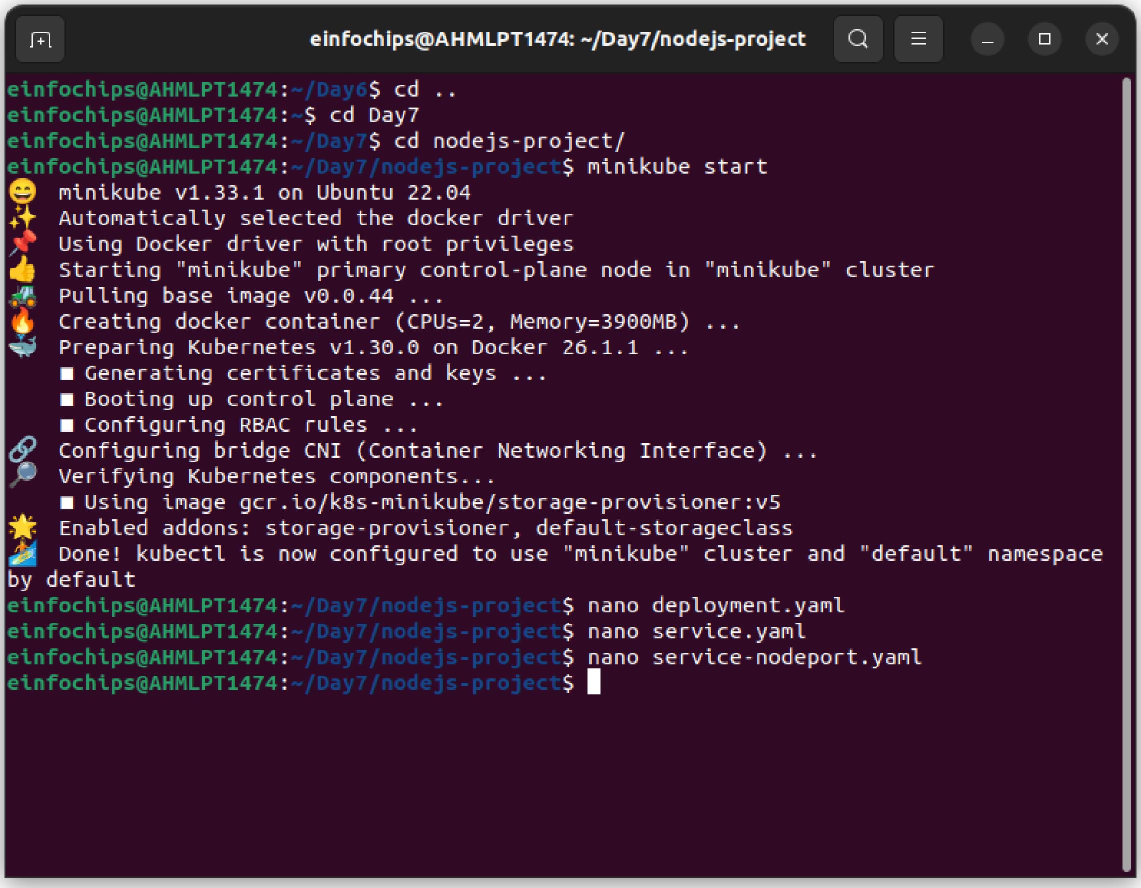
Create a `service.yaml` file for ClusterIP:

```
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
```



```
einfochips@AHMLPT1474:~/Day6$ cd ..
einfochips@AHMLPT1474:~$ cd Day7
einfochips@AHMLPT1474:~/Day7$ cd nodejs-project/
einfochips@AHMLPT1474:~/Day7/nodejs-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04
✨  Automatically selected the docker driver
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔥  Creating docker container (CPUs=2, Memory=3900MB) ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace
by default
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano deployment.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service-nodeport.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ 
```

4.3. Apply Manifests to Minikube

Apply the deployment:

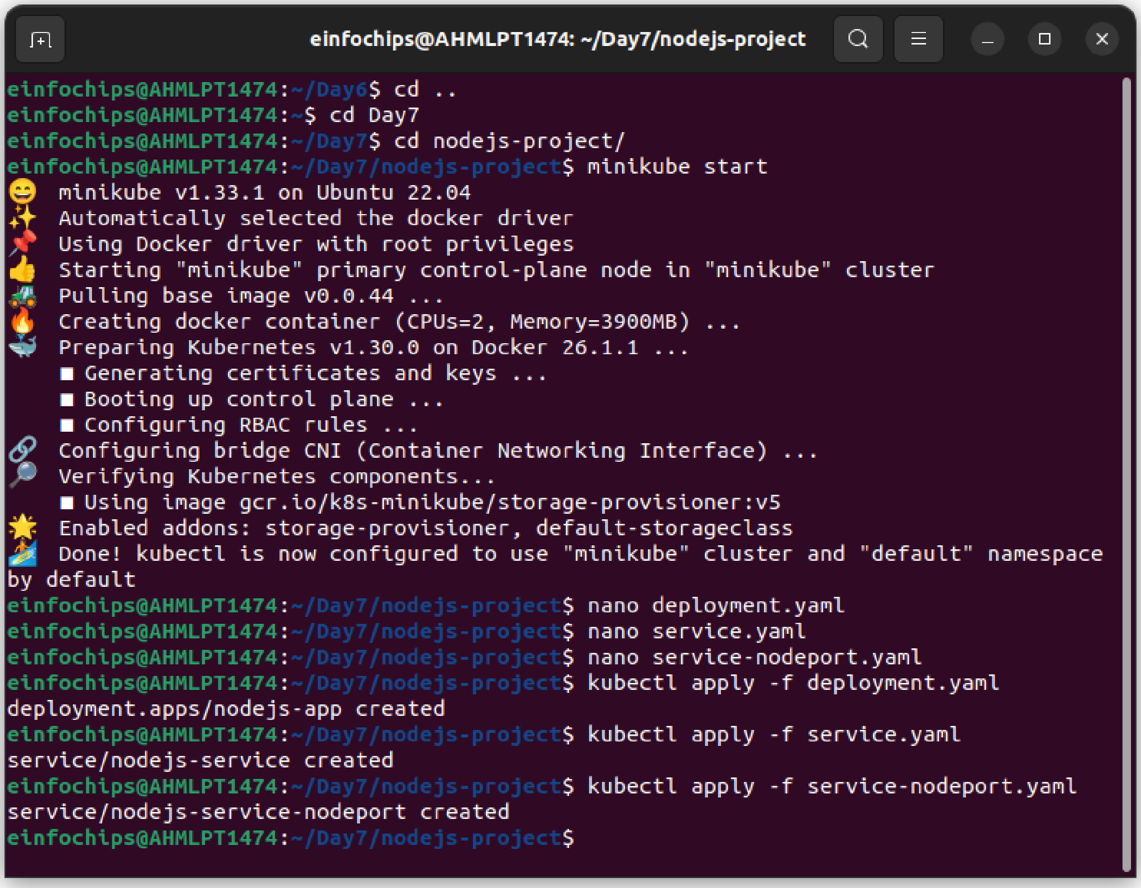
```
kubectl apply -f deployment.yaml
```

Apply the ClusterIP service:

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

Apply the NodePort service:

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



einfochips@AHMLPT1474: ~/Day7/nodejs-project

```
einfochips@AHMLPT1474:~/Day6$ cd ..
einfochips@AHMLPT1474:~$ cd Day7
einfochips@AHMLPT1474:~/Day7$ cd nodejs-project/
einfochips@AHMLPT1474:~/Day7/nodejs-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04
✨  Automatically selected the docker driver
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔥  Creating docker container (CPUs=2, Memory=3900MB) ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace
by default
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano deployment.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service-nodeport.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app created
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f service.yaml
service/nodejs-service created
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f service-nodeport.yaml
service/nodejs-service-nodeport created
einfochips@AHMLPT1474:~/Day7/nodejs-project$
```

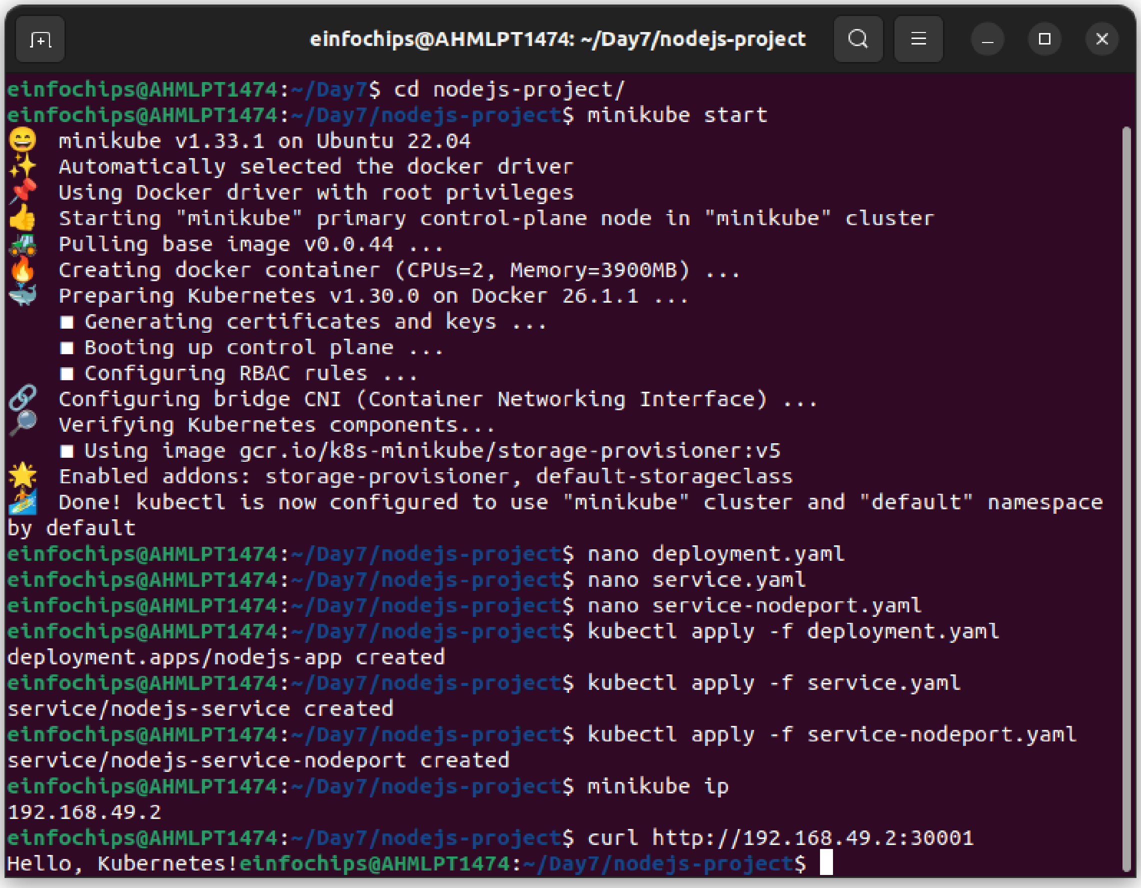
4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

1. Access the application using the NodePort:

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



```
einfochips@AHMLPT1474:~/Day7$ cd nodejs-project/
einfochips@AHMLPT1474:~/Day7/nodejs-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04
✨  Automatically selected the docker driver
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔥  Creating docker container (CPUs=2, Memory=3900MB) ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace
by default
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano deployment.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano service-nodeport.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app created
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f service.yaml
service/nodejs-service created
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f service-nodeport.yaml
service/nodejs-service-nodeport created
einfochips@AHMLPT1474:~/Day7/nodejs-project$ minikube ip
192.168.49.2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ curl http://192.168.49.2:30001
Hello, Kubernetes!einfochips@AHMLPT1474:~/Day7/nodejs-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"
```

7. Merge the Changes and Rebuild the Docker Image

7.1. Merge the Feature Branch

Switch back to the `main` branch:

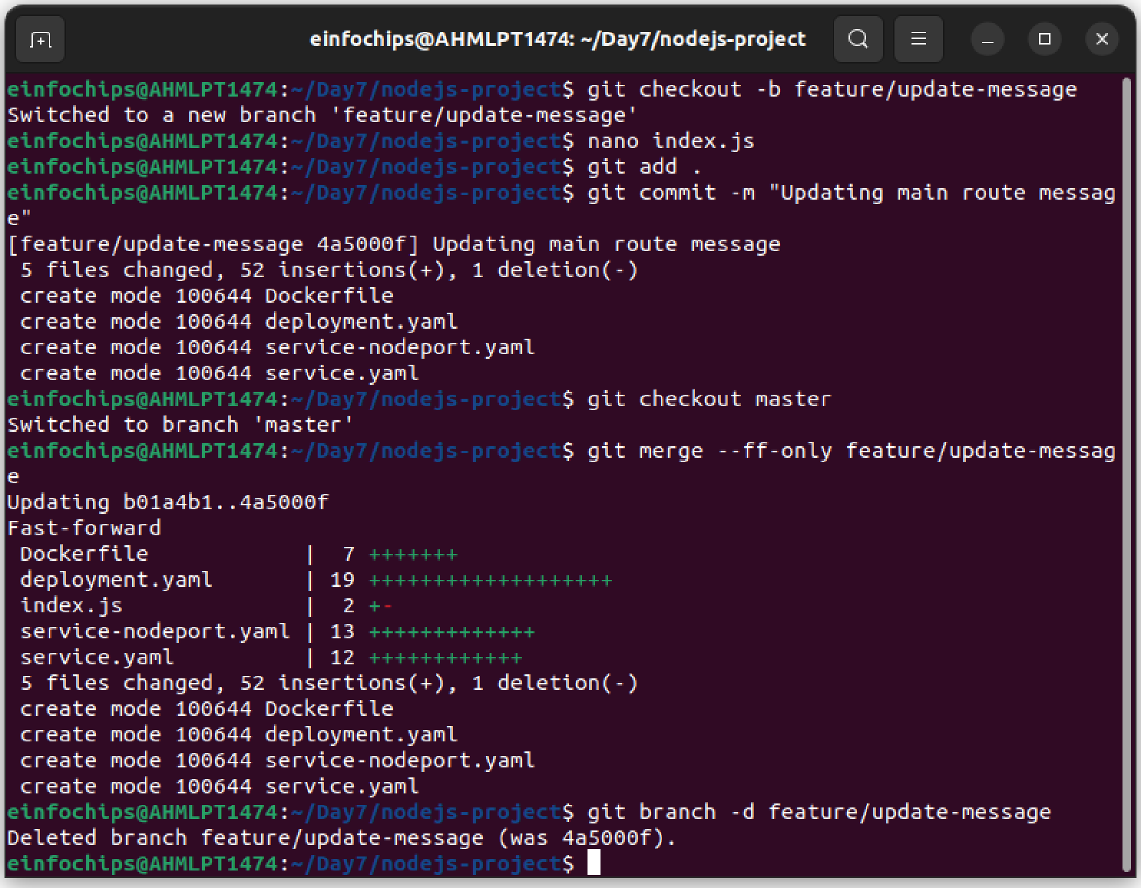
```
git checkout main
```

Merge the `feature/update-message` branch:

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

Delete the feature branch:

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

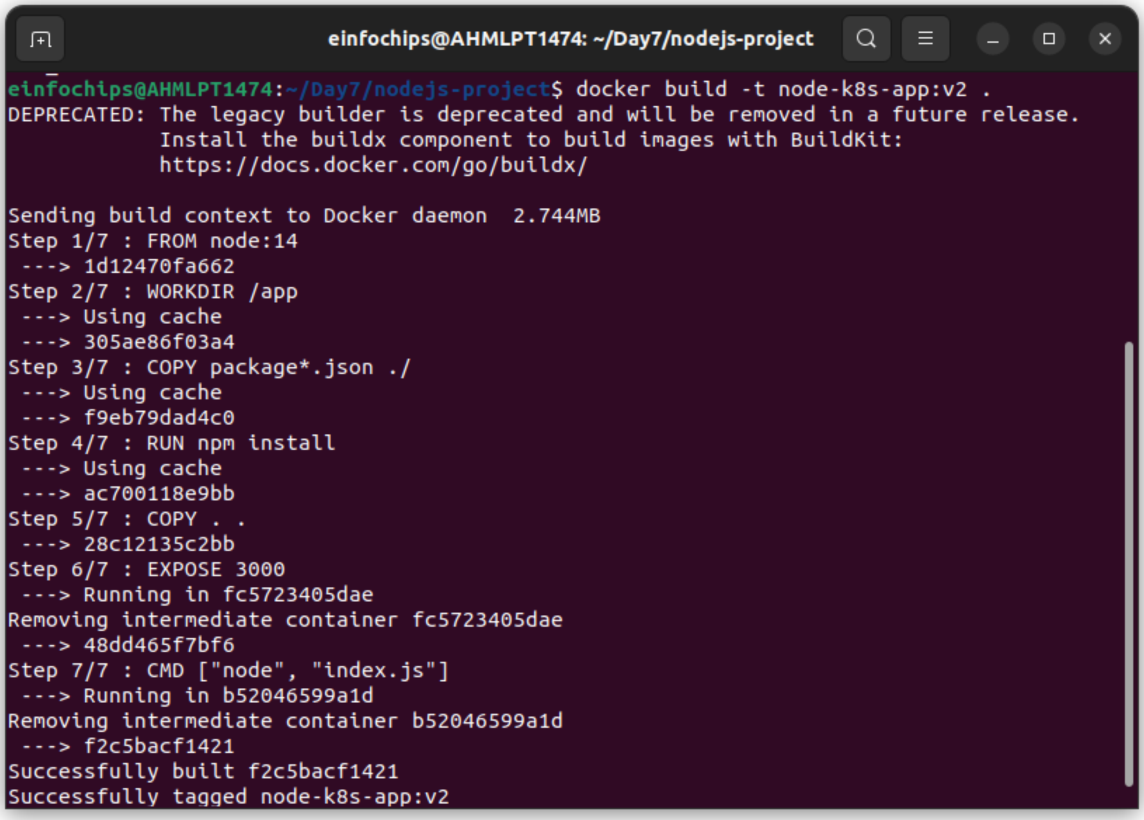


```
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git checkout -b feature/update-message
Switched to a new branch 'feature/update-message'
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano index.js
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git add .
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git commit -m "Updating main route messag
e"
[feature/update-message 4a5000f] Updating main route message
 5 files changed, 52 insertions(+), 1 deletion(-)
 create mode 100644 Dockerfile
 create mode 100644 deployment.yaml
 create mode 100644 service-nodeport.yaml
 create mode 100644 service.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git checkout master
Switched to branch 'master'
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git merge --ff-only feature/update-messag
e
Updating b01a4b1..4a5000f
Fast-forward
 Dockerfile            |  7 +++++++
 deployment.yaml       | 19 +++++++++++++++++++
 index.js              |  2 +-
 service-nodeport.yaml | 13 +++++++++++++
 service.yaml          | 12 ++++++++++++
 5 files changed, 52 insertions(+), 1 deletion(-)
 create mode 100644 Dockerfile
 create mode 100644 deployment.yaml
 create mode 100644 service-nodeport.yaml
 create mode 100644 service.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ git branch -d feature/update-message
Deleted branch feature/update-message (was 4a5000f).
einfochips@AHMLPT1474:~/Day7/nodejs-project$
```

7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

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



```
einfochips@AHMLPT1474:~/Day7/nodejs-project$ docker build -t node-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  2.744MB
Step 1/7 : FROM node:14
 ---> 1d12470fa662
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 305ae86f03a4
Step 3/7 : COPY package*.json ./
 ---> Using cache
 ---> f9eb79dad4c0
Step 4/7 : RUN npm install
 ---> Using cache
 ---> ac700118e9bb
Step 5/7 : COPY . .
 ---> 28c12135c2bb
Step 6/7 : EXPOSE 3000
 ---> Running in fc5723405dae
Removing intermediate container fc5723405dae
 ---> 48dd465f7bf6
Step 7/7 : CMD ["node", "index.js"]
 ---> Running in b52046599a1d
Removing intermediate container b52046599a1d
 ---> f2c5bacf1421
Successfully built f2c5bacf1421
Successfully tagged node-k8s-app:v2
```

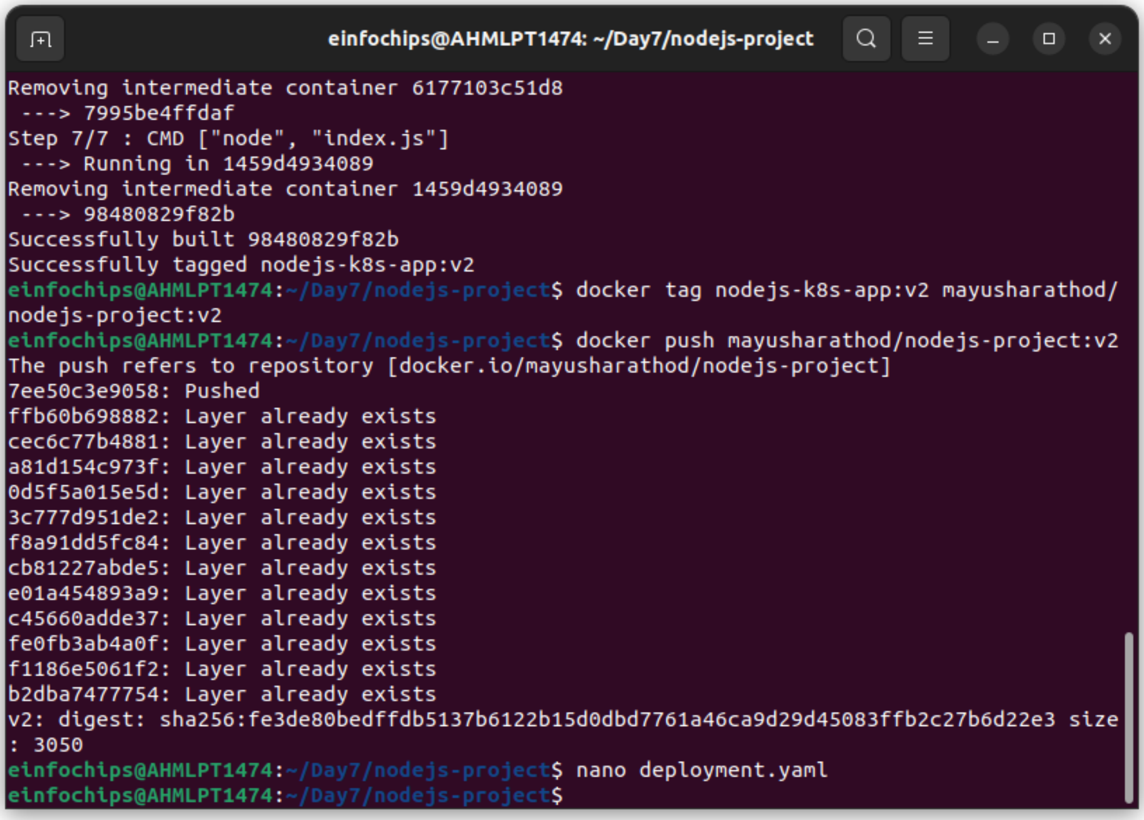
8. Update Kubernetes Deployment

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
```



```
Removing intermediate container 6177103c51d8
 ---> 7995be4ffdaf
Step 7/7 : CMD ["node", "index.js"]
 ---> Running in 1459d4934089
Removing intermediate container 1459d4934089
 ---> 98480829f82b
Successfully built 98480829f82b
Successfully tagged nodejs-k8s-app:v2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ docker tag nodejs-k8s-app:v2 mayusharathod/
nodejs-project:v2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ docker push mayusharathod/nodejs-project:v2
The push refers to repository [docker.io/mayusharathod/nodejs-project]
7ee50c3e9058: Pushed
ffb60b698882: Layer already exists
cec6c77b4881: Layer already exists
a81d154c973f: 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:fe3de80bedffdb5137b6122b15d0dbd7761a46ca9d29d45083ffb2c27b6d22e3 size
: 3050
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano deployment.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$
```

8.2. Apply the Updated Manifest

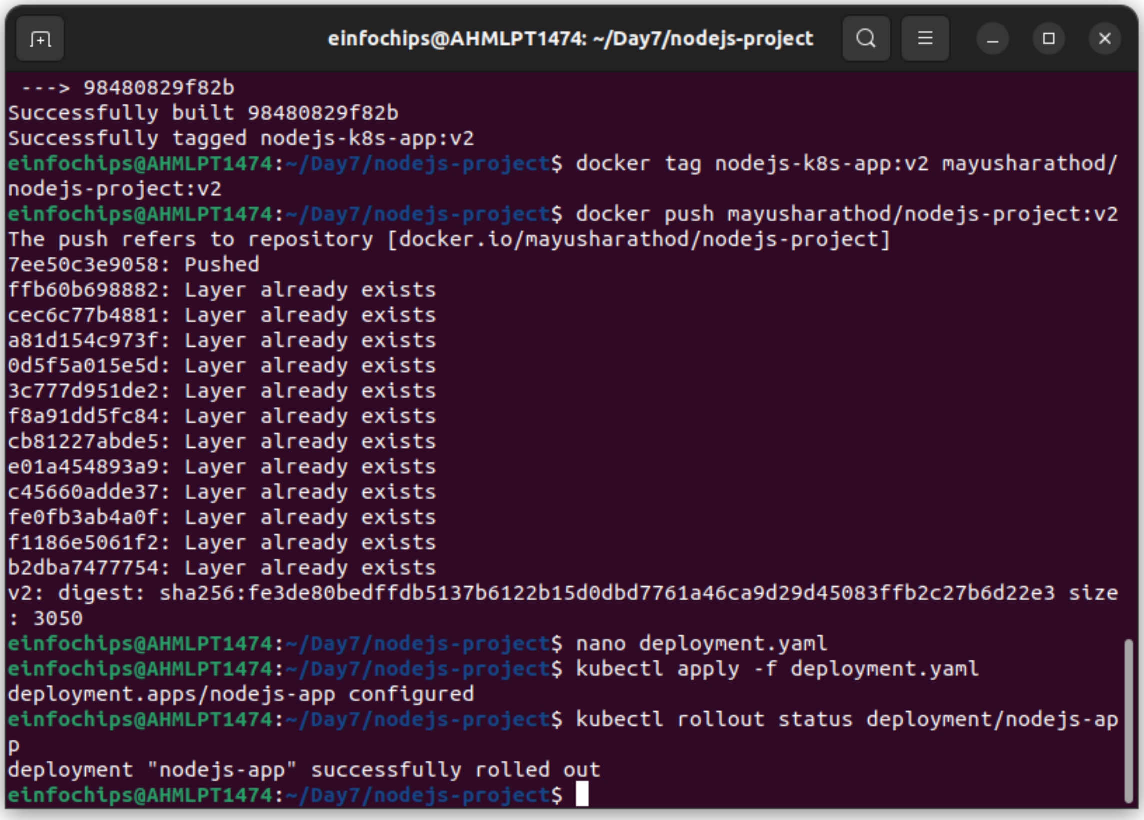
Apply the updated deployment:

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

8.3. Verify the Update

Check the status of the deployment:

kubectl rollout status deployment/nodejs-app



```
 ---> 98480829f82b
Successfully built 98480829f82b
Successfully tagged nodejs-k8s-app:v2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ docker tag nodejs-k8s-app:v2 mayusharathod/
nodejs-project:v2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ docker push mayusharathod/nodejs-project:v2
The push refers to repository [docker.io/mayusharathod/nodejs-project]
7ee50c3e9058: Pushed
ffb60b698882: Layer already exists
cec6c77b4881: Layer already exists
a81d154c973f: 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:fe3de80bedffdb5137b6122b15d0dbd7761a46ca9d29d45083ffb2c27b6d22e3 size
: 3050
einfochips@AHMLPT1474:~/Day7/nodejs-project$ nano deployment.yaml
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app configured
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl rollout status deployment/nodejs-ap
p
deployment "nodejs-app" successfully rolled out
einfochips@AHMLPT1474:~/Day7/nodejs-project$ 
```

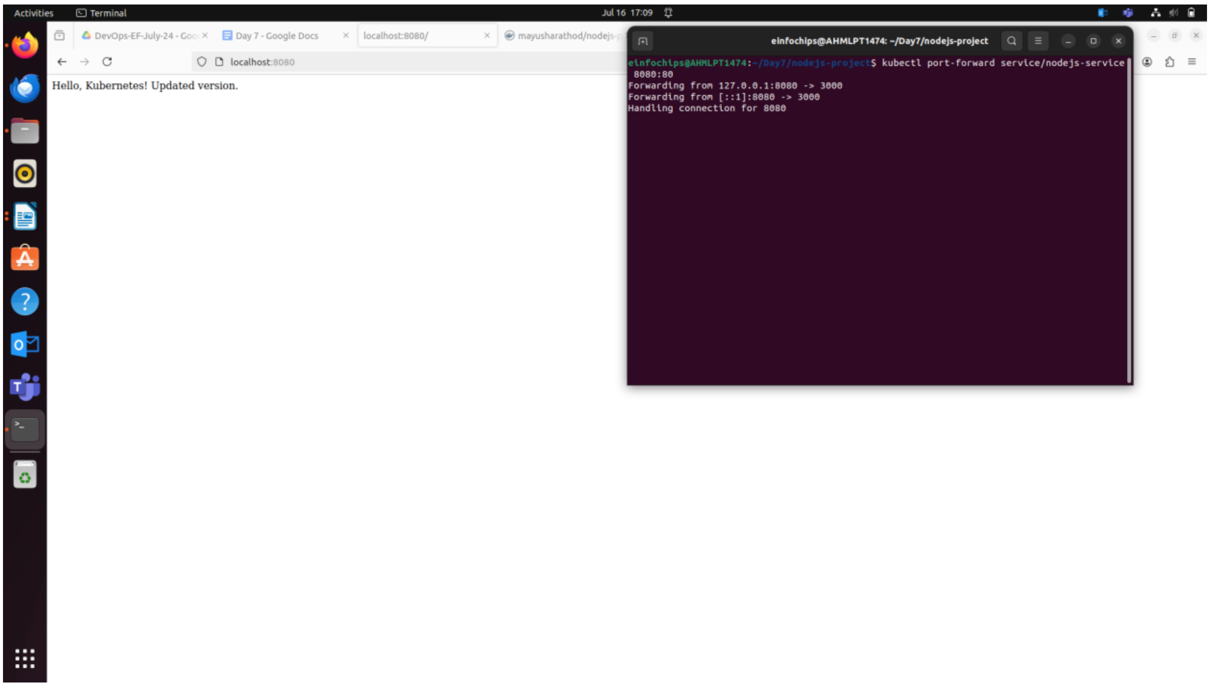
9. Access the Updated Application

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.



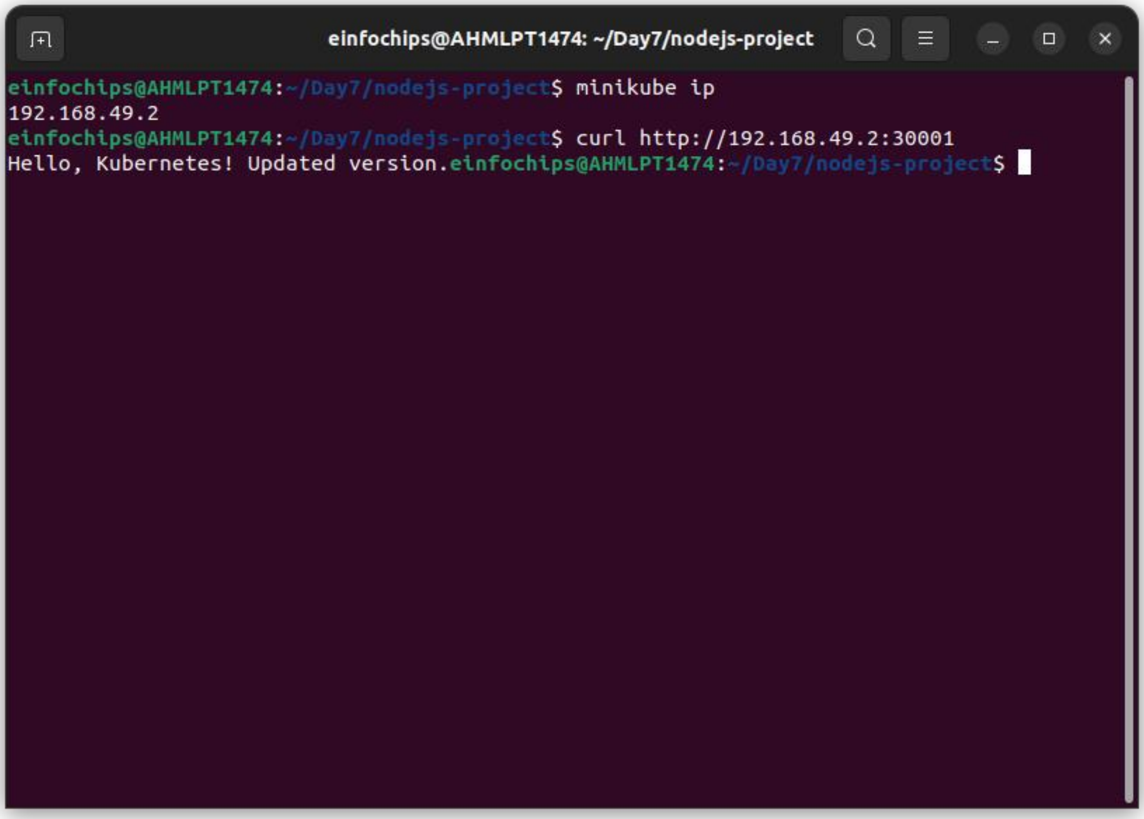
Hello, Kubernetes! Updated version.

```
einfochips@AHMLPT1474:~/Day7/nodejs-project$ kubectl port-forward service/nodejs-service
 8080:80
Forwarding from 127.0.0.1:8080 -> 3000
Forwarding from [::1]:8080 -> 3000
Handling connection for 8080
```

9.2. Access Through NodePort Service

1. Access the application using the NodePort:

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



```
einfochips@AHMLPT1474:~/Day7/nodejs-project$ minikube ip
192.168.49.2
einfochips@AHMLPT1474:~/Day7/nodejs-project$ curl http://192.168.49.2:30001
Hello, Kubernetes! Updated version.einfochips@AHMLPT1474:~/Day7/nodejs-project$ 
```