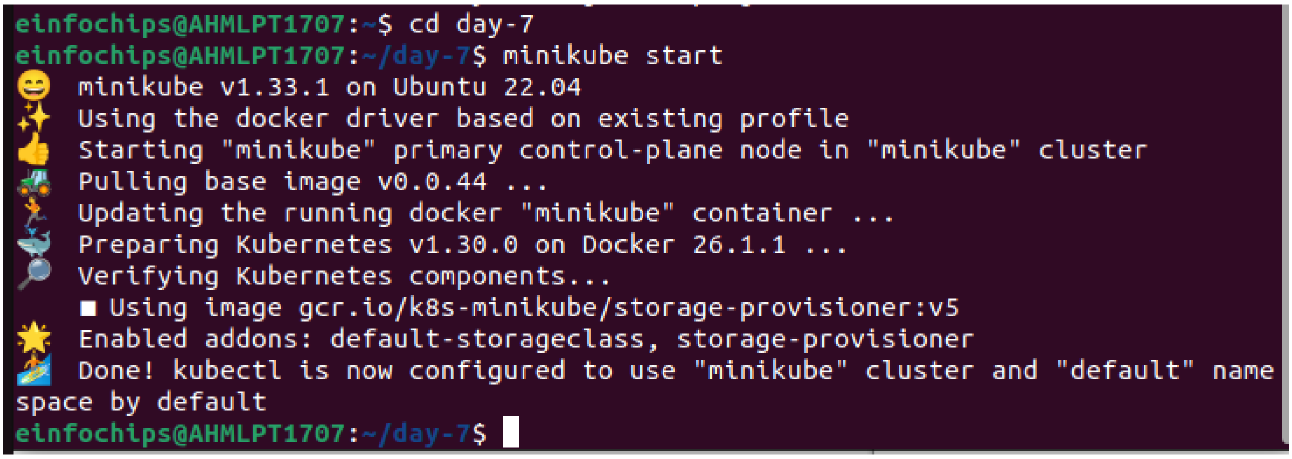
In this project, you will develop a simple Node.js application, deploy it on a local Kubernetes cluster using Minikube, and configure various Kubernetes features. The project includes Git version control practices, creating and managing branches, and performing rebases. Additionally, you will work with ConfigMaps, Secrets, environment variables, and set up vertical and horizontal pod autoscaling.

Setup Minikube and Git Repository

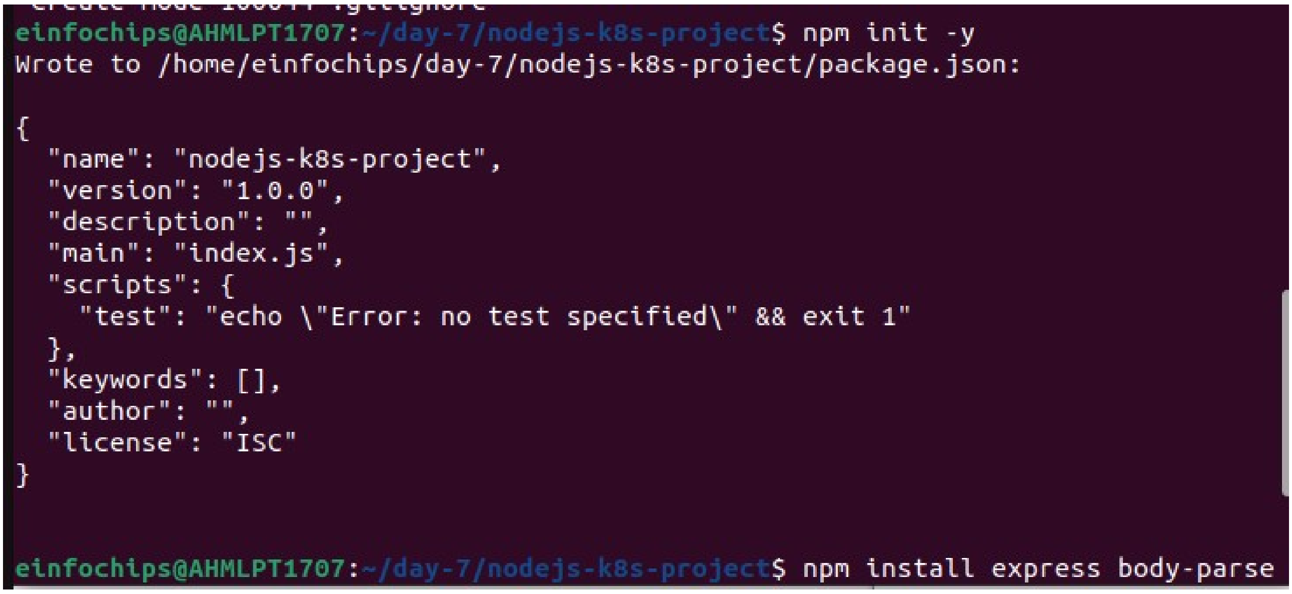


```
einfochips@AHMLPT1707:~$ cd day-7
einfochips@AHMLPT1707:~/day-7$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🏃  Updating the running docker "minikube" container ...
🐳  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@AHMLPT1707:~/day-7$ 
```

Develop a Node.js Application

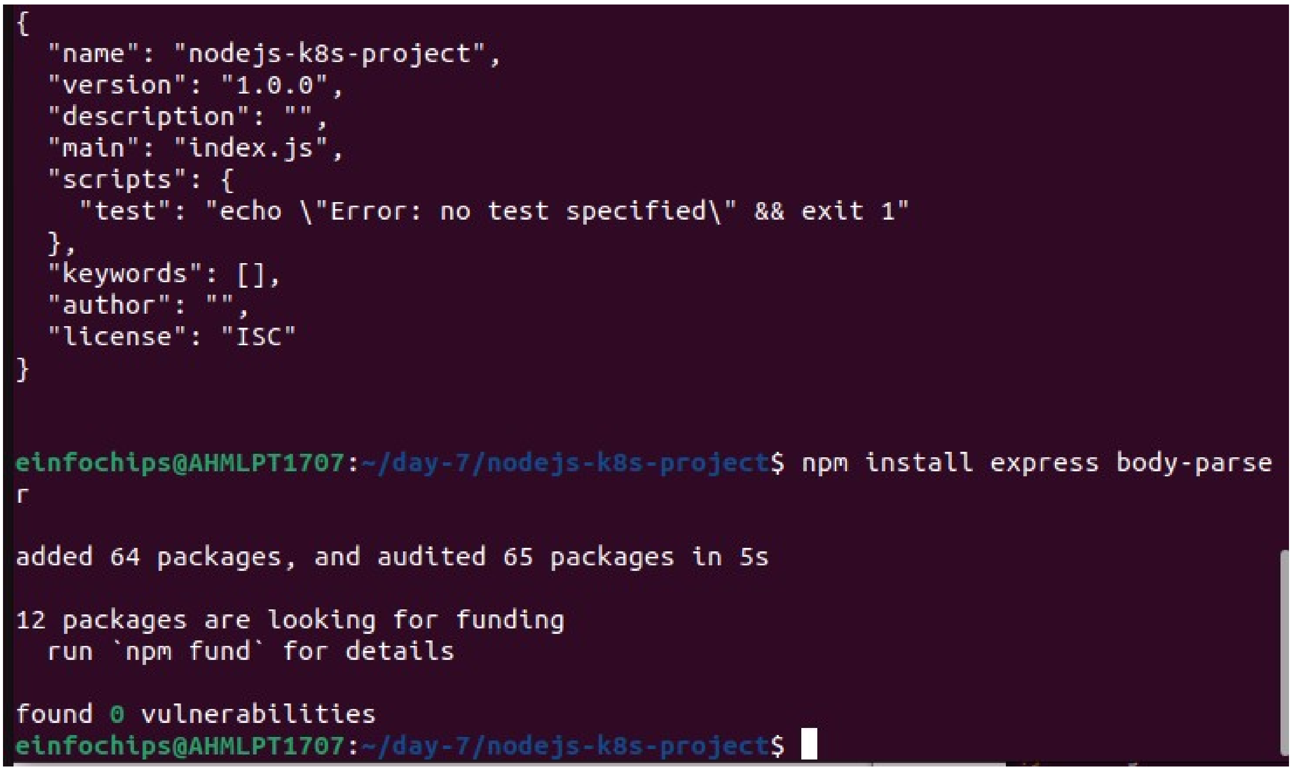
Create the Node.js App

Initialize the Node.js project:



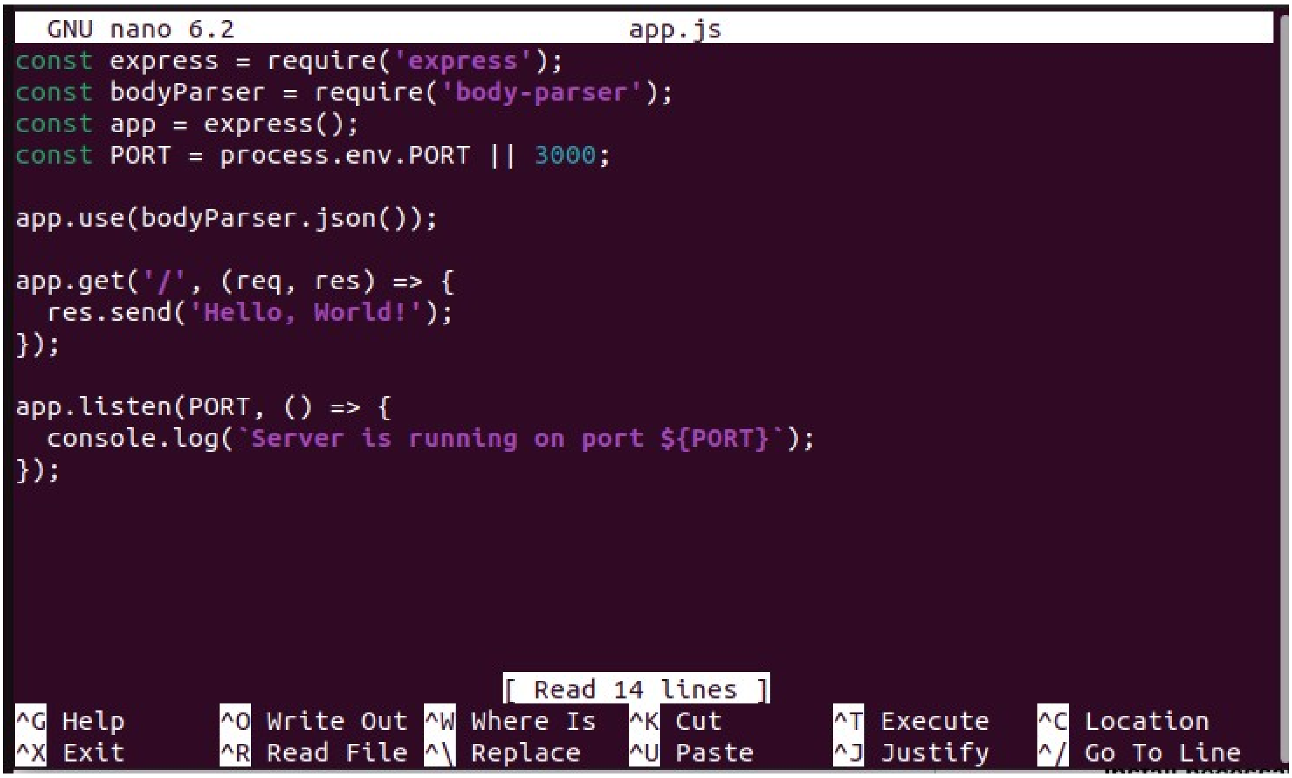
```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ npm init -y
Wrote to /home/einfochips/day-7/nodejs-k8s-project/package.json:

{
  "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"
}


einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ npm install express body-parse
```

Install necessary packages:



```
{
  "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@AHMLPT1707:~/day-7/nodejs-k8s-project$ npm install express body-parse
r

added 64 packages, and audited 65 packages in 5s

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

found 0 vulnerabilities
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ 
```

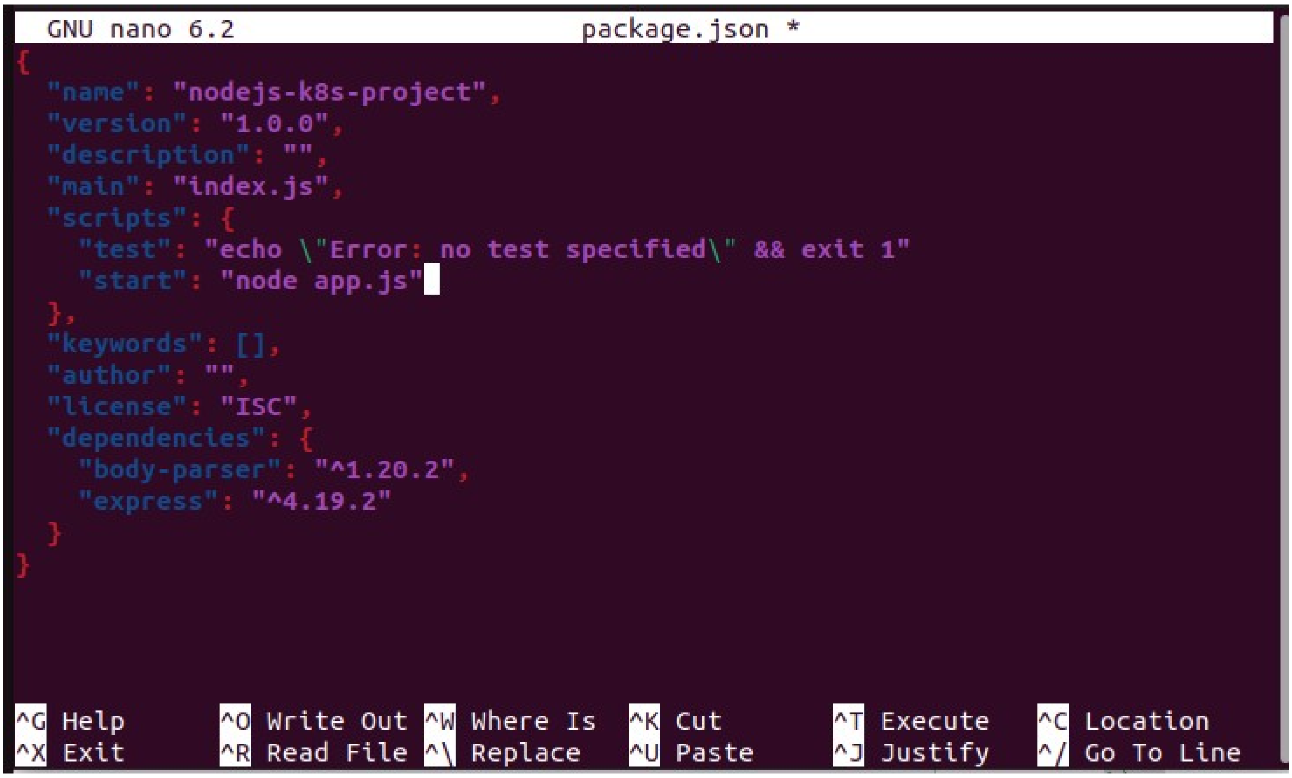
create app.js



```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const PORT = process.env.PORT || 3000;

app.use(bodyParser.json());

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

```
[ Read 14 lines ]
```

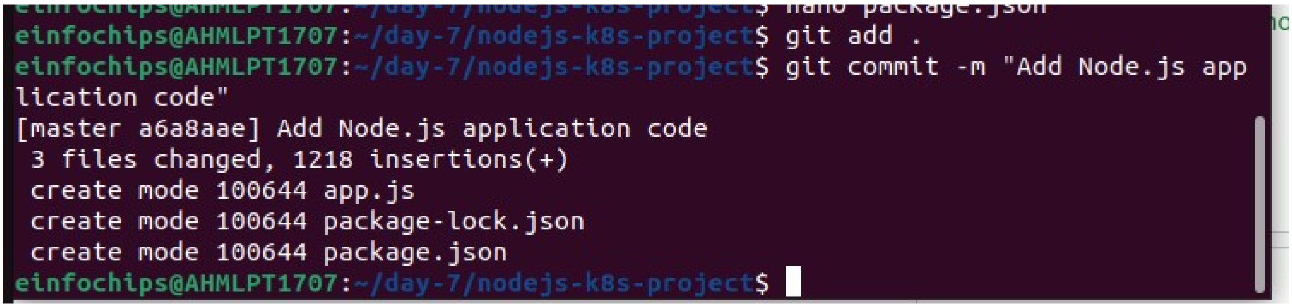
Update package.json



```
{
  "name": "nodejs-k8s-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node app.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "express": "^4.19.2"
  }
}
```

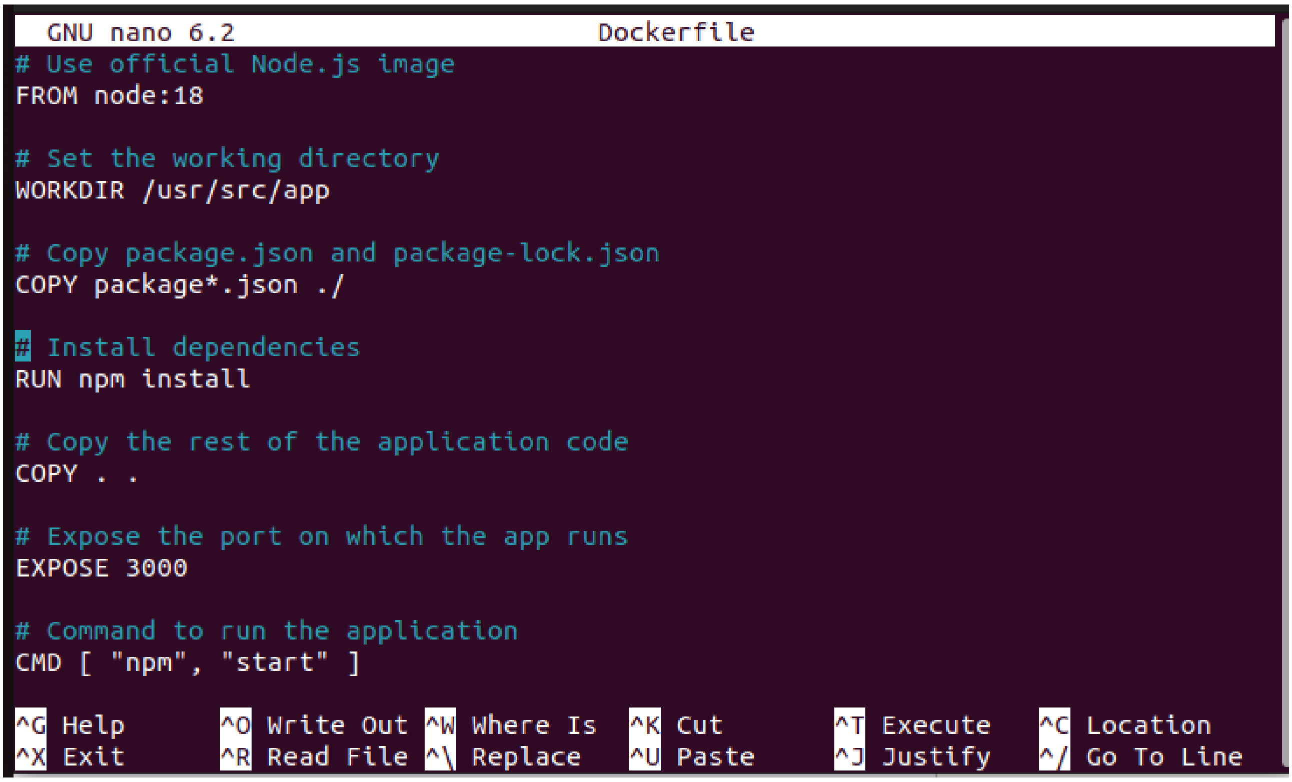
Commit the Node.js Application

Add and commit changes:



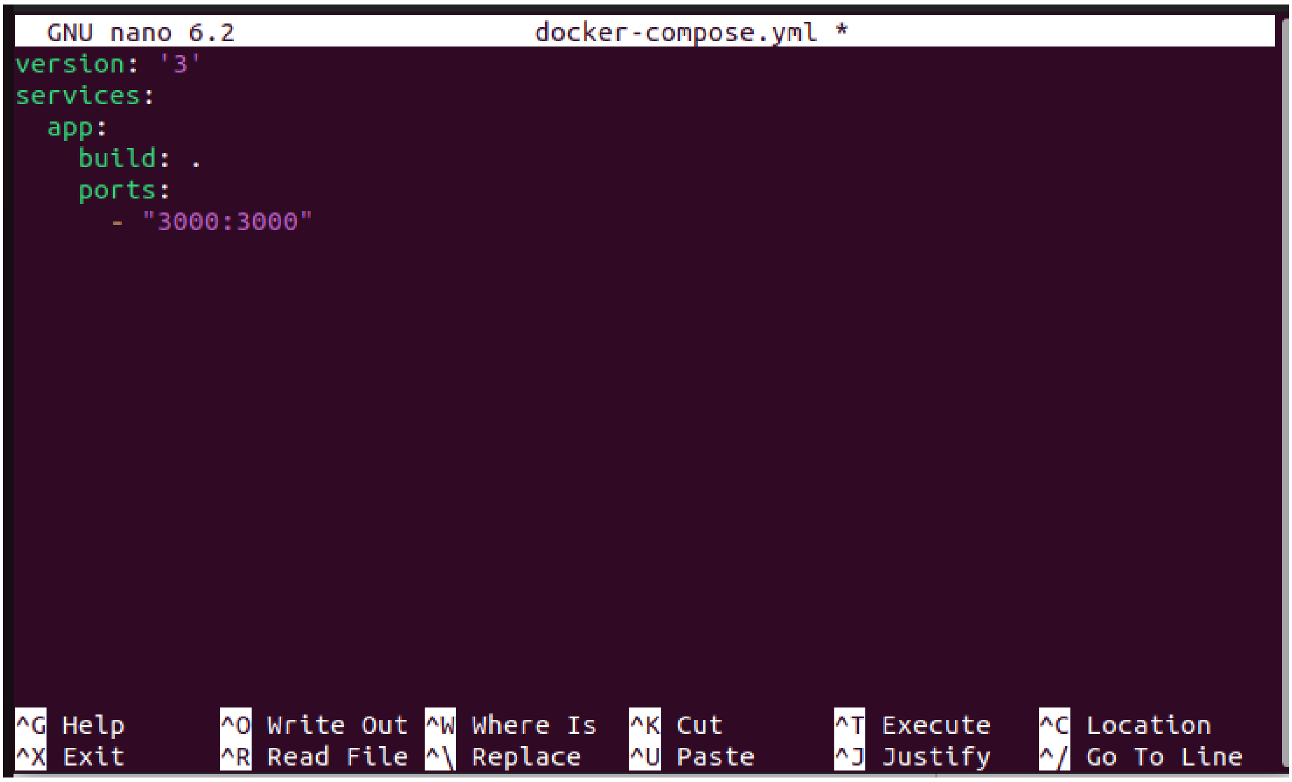
```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano package.json
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git commit -m "Add Node.js app
lication code"
[master a6a8aae] Add Node.js application code
 3 files changed, 1218 insertions(+)
 create mode 100644 app.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ 
```

Create Dockerfile and Docker Compose



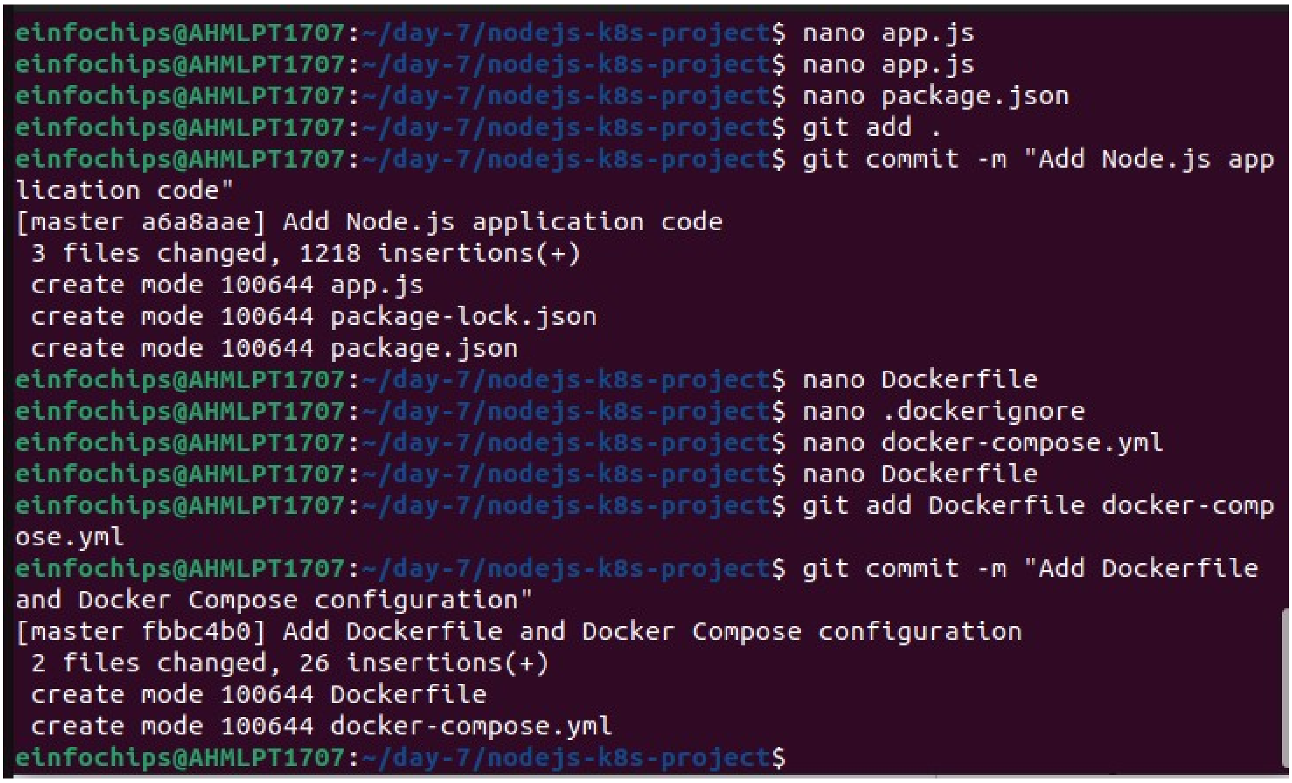
```
  GNU nano 6.2                         Dockerfile
# Use official Node.js image
FROM node:18

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

# Copy package.json and package-lock.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code
COPY . .

# Expose the port on which the app runs
EXPOSE 3000

# Command to run the application
CMD [ "npm", "start" ]
```

Create docker-compose.yml (optional for local testing)



```
  GNU nano 6.2                    docker-compose.yml *
version: '3'
services:
  app:
    build: .
    ports:
      - "3000:3000"
```

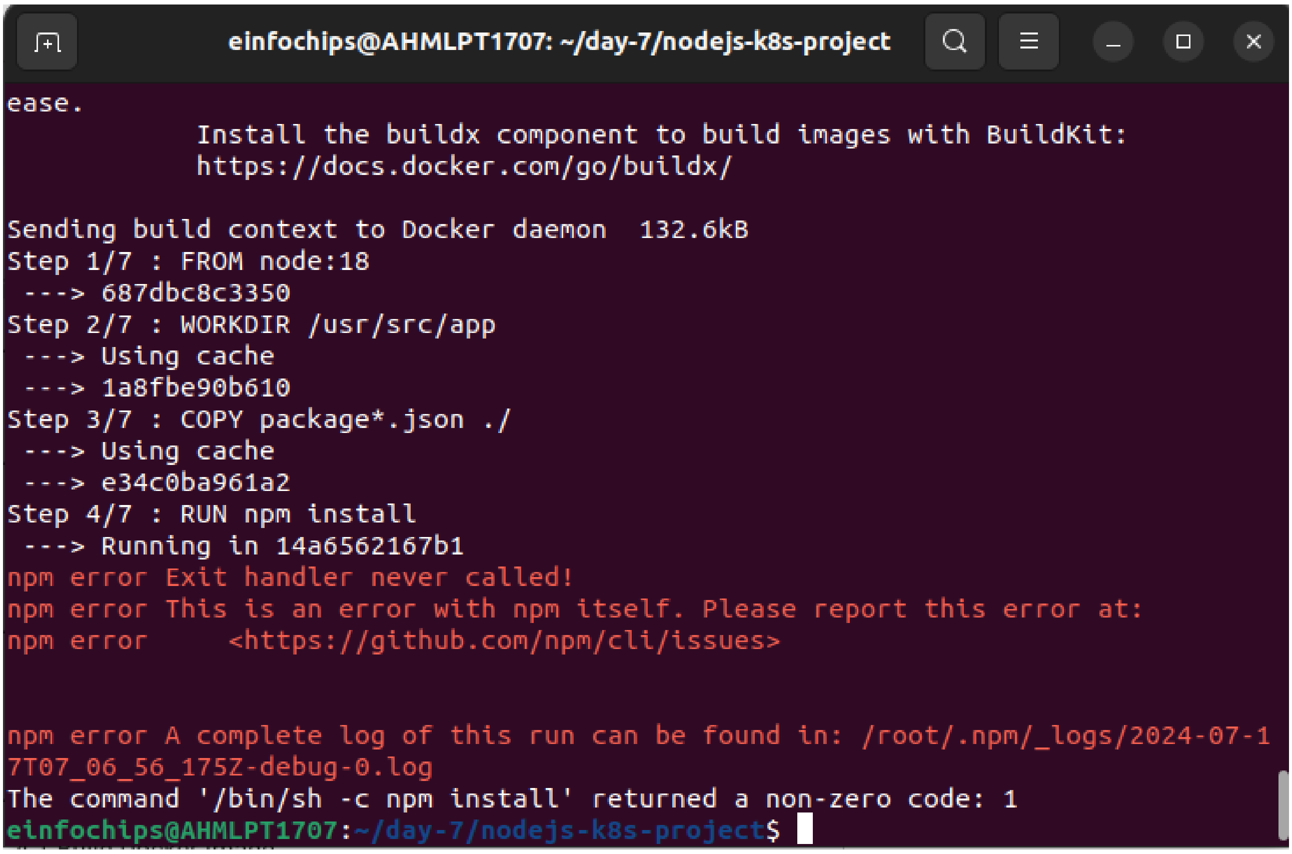
Add and commit changes:



```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano app.js
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano app.js
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano package.json
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git add .
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git commit -m "Add Node.js app
lication code"
[master a6a8aae] Add Node.js application code
 3 files changed, 1218 insertions(+)
 create mode 100644 app.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano Dockerfile
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano .dockerignore
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano docker-compose.yml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano Dockerfile
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git add Dockerfile docker-comp
ose.yml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git commit -m "Add Dockerfile
and Docker Compose configuration"
[master fbbc4b0] Add Dockerfile and Docker Compose configuration
 2 files changed, 26 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 docker-compose.yml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

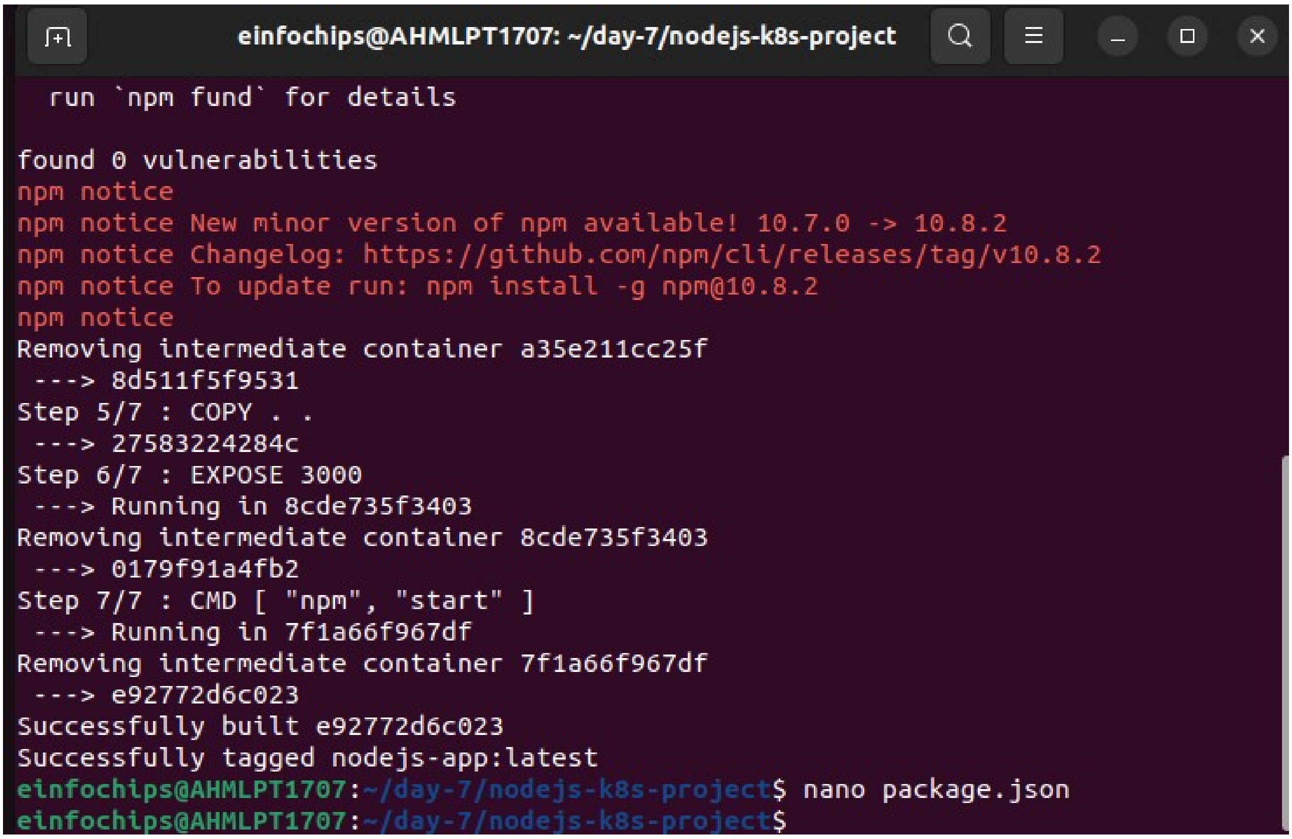
Build and Push Docker Image

FACED ISSUE IN THIS AND THE PROCESS IS STILL STUCK HERE



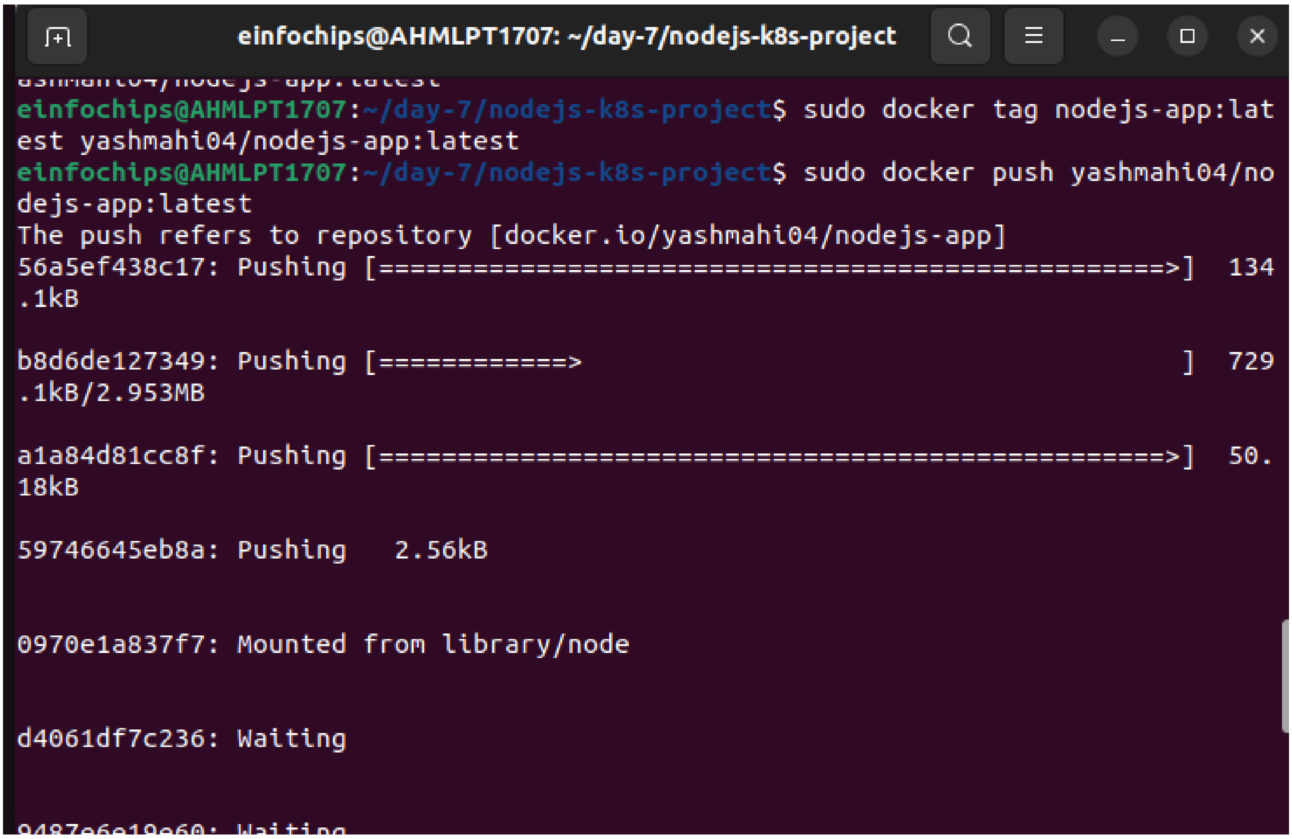
```
ease.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  132.6kB
Step 1/7 : FROM node:18
 ---> 687dbc8c3350
Step 2/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> 1a8fbe90b610
Step 3/7 : COPY package*.json ./
 ---> Using cache
 ---> e34c0ba961a2
Step 4/7 : RUN npm install
 ---> Running in 14a6562167b1
npm error Exit handler never called!
npm error This is an error with npm itself. Please report this error at:
npm error     <https://github.com/npm/cli/issues>


npm error A complete log of this run can be found in: /root/.npm/_logs/2024-07-17T07_06_56_175Z-debug-0.log
The command '/bin/sh -c npm install' returned a non-zero code: 1
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ 
```

Solved the issue which was in package.json under start: node-app.js previously it was node app.js



```
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice
Removing intermediate container a35e211cc25f
 ---> 8d511f5f9531
Step 5/7 : COPY . .
 ---> 27583224284c
Step 6/7 : EXPOSE 3000
 ---> Running in 8cde735f3403
Removing intermediate container 8cde735f3403
 ---> 0179f91a4fb2
Step 7/7 : CMD [ "npm", "start" ]
 ---> Running in 7f1a66f967df
Removing intermediate container 7f1a66f967df
 ---> e92772d6c023
Successfully built e92772d6c023
Successfully tagged nodejs-app:latest
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano package.json
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

Push Docker Image to Docker Hub

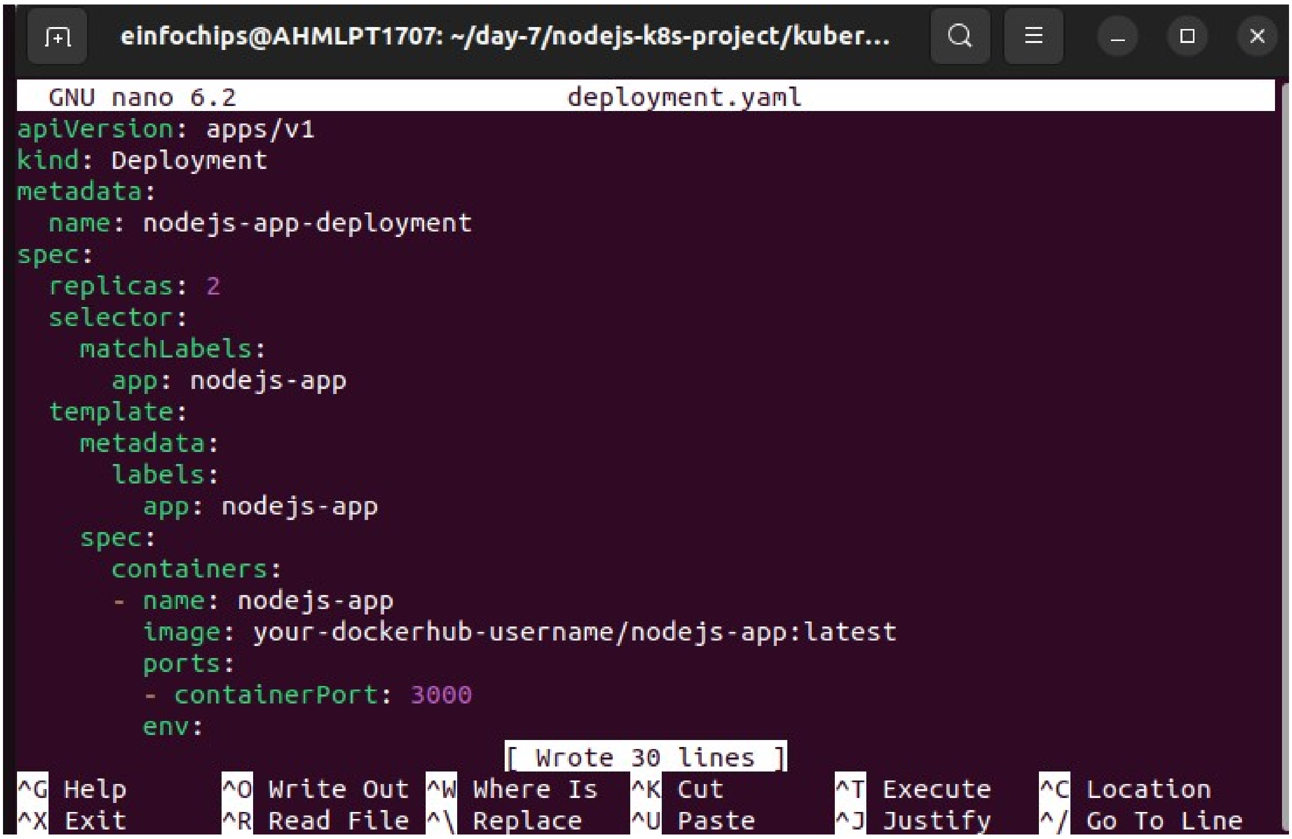


```
ashmahi04/nodejs-app:latest
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ sudo docker tag nodejs-app:lat
est yashmahi04/nodejs-app:latest
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ sudo docker push yashmahi04/no
dejs-app:latest
The push refers to repository [docker.io/yashmahi04/nodejs-app]
56a5ef438c17: Pushing [==================================================>]  134
.1kB

b8d6de127349: Pushing [============>                                      ]  729
.1kB/2.953MB

a1a84d81cc8f: Pushing [==================================================>]  50.
18kB

59746645eb8a: Pushing   2.56kB


0970e1a837f7: Mounted from library/node


d4061df7c236: Waiting


9487e6e19e60: Waiting
```

Add and commit changes

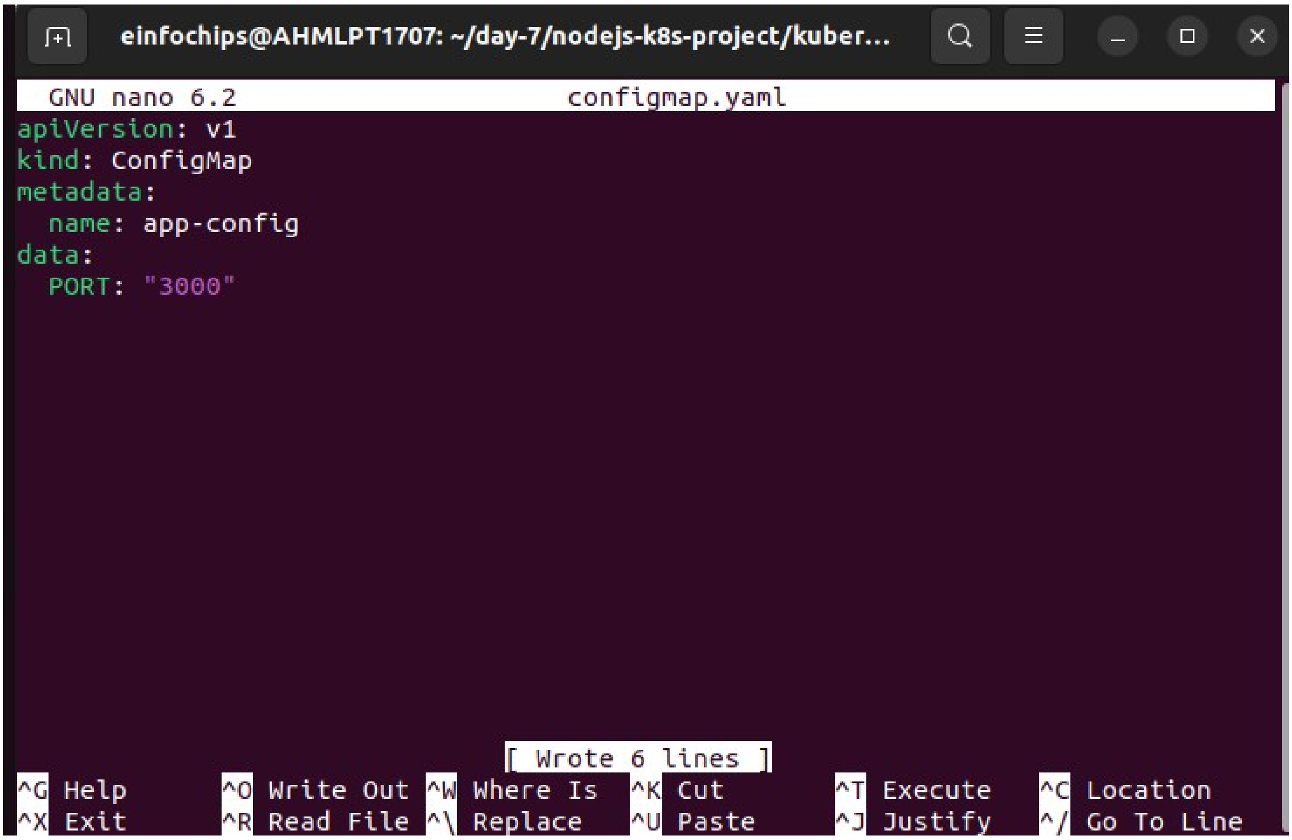
Create Kubernetes Configurations

Create kubernetes/deployment.yaml:



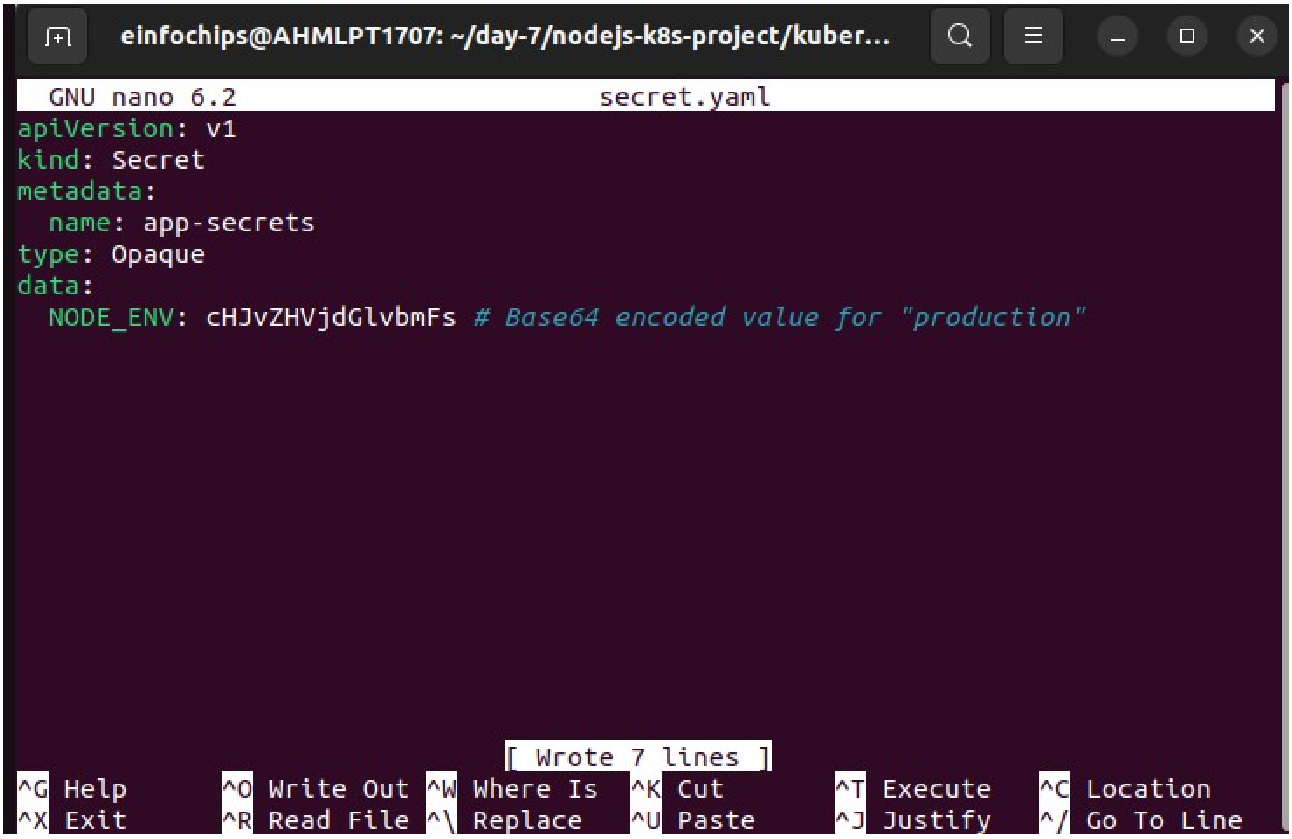
```
  GNU nano 6.2                        deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - name: nodejs-app
        image: your-dockerhub-username/nodejs-app:latest
        ports:
        - containerPort: 3000
        env:
                               [ Wrote 30 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

kubernetes/config.yaml



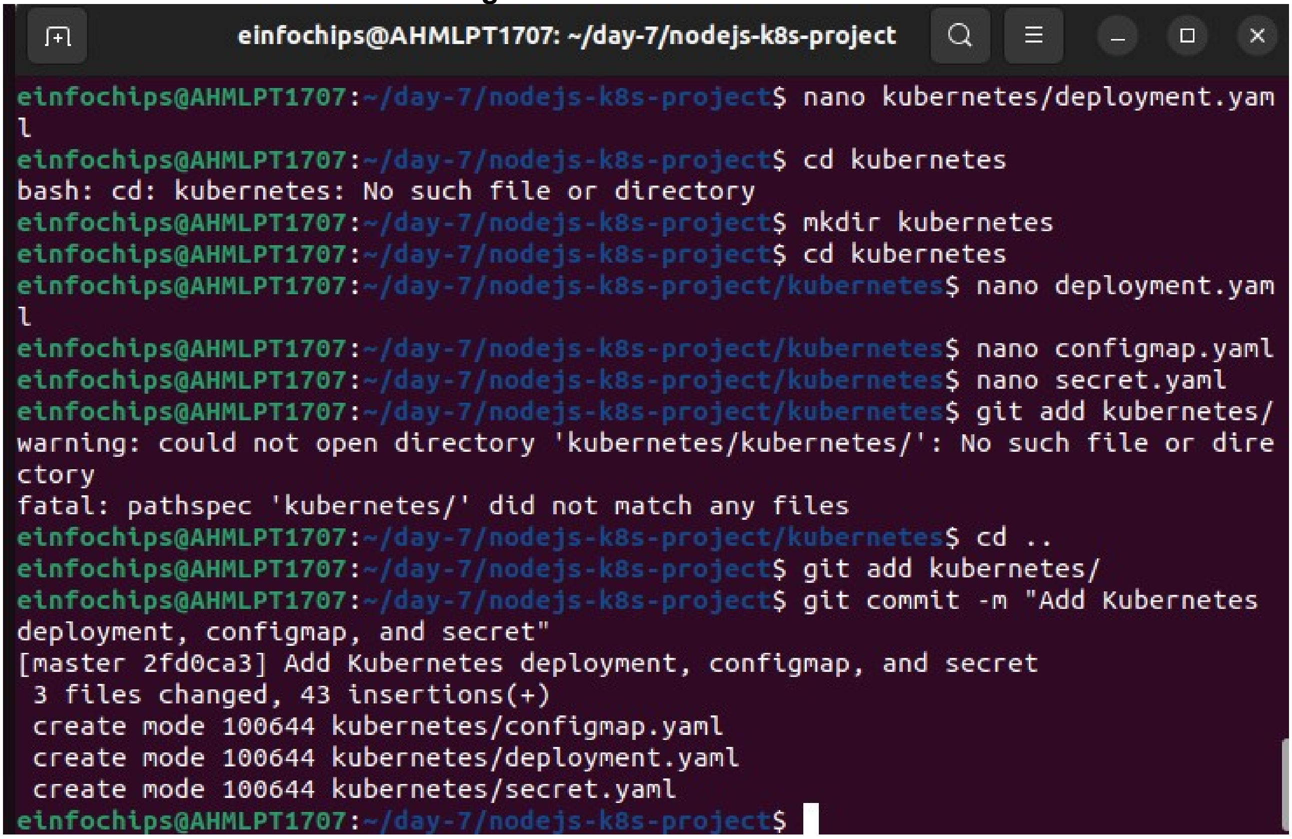
```
  GNU nano 6.2                        configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"
                               [ Wrote 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

kubernetes/secret.yaml



```
  GNU nano 6.2                          secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbmFs # Base64 encoded value for "production"




                                [ Wrote 7 lines ]
^G Help        ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```

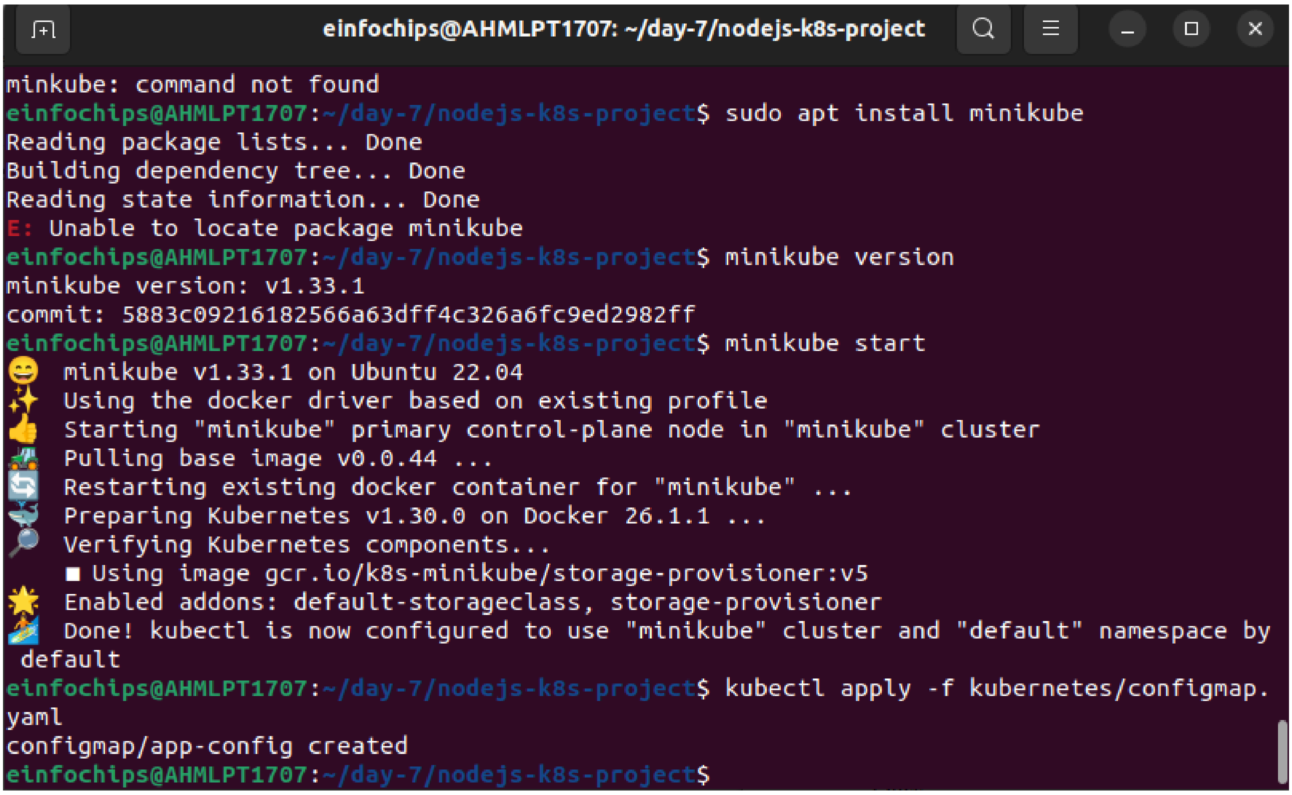
Add and commit Kubernetes configurations:



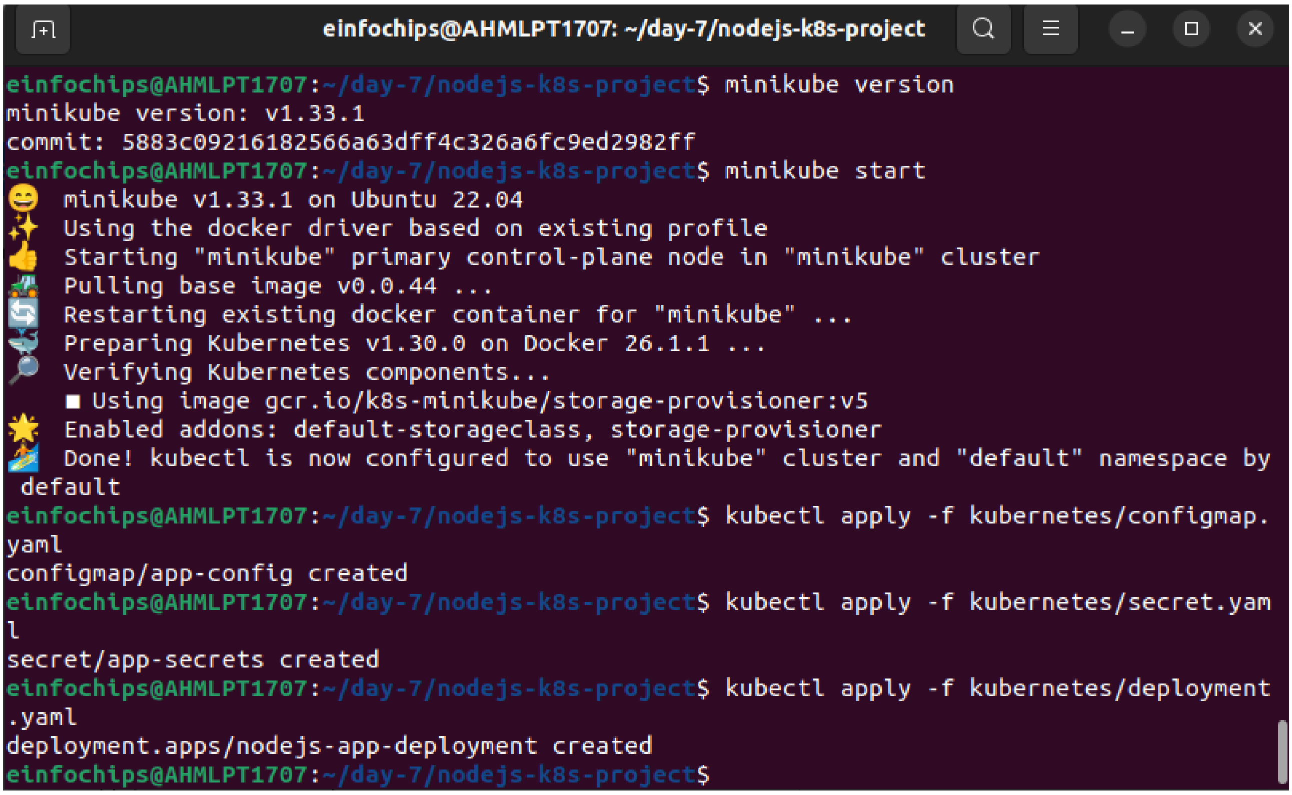
```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ nano kubernetes/deployment.yam
l
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
bash: cd: kubernetes: No such file or directory
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ mkdir kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano deployment.yam
l
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano configmap.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano secret.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ git add kubernetes/
warning: could not open directory 'kubernetes/kubernetes/': No such file or dire
ctory
fatal: pathspec 'kubernetes/' did not match any files
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git add kubernetes/
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ git commit -m "Add Kubernetes
deployment, configmap, and secret"
[master 2fd0ca3] Add Kubernetes deployment, configmap, and secret
 3 files changed, 43 insertions(+)
 create mode 100644 kubernetes/configmap.yaml
 create mode 100644 kubernetes/deployment.yaml
 create mode 100644 kubernetes/secret.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ 
```

Apply Kubernetes Configurations

Apply the ConfigMap and Secret:



```
minikube: command not found
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ sudo apt install minikube
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package minikube
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ minikube version
minikube version: v1.33.1
commit: 5883c09216182566a63dff4c326a6fc9ed2982ff
einfochips@AHMLPT1707:~/day-7/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" namespace by
 default
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/configmap.
yaml
configmap/app-config created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

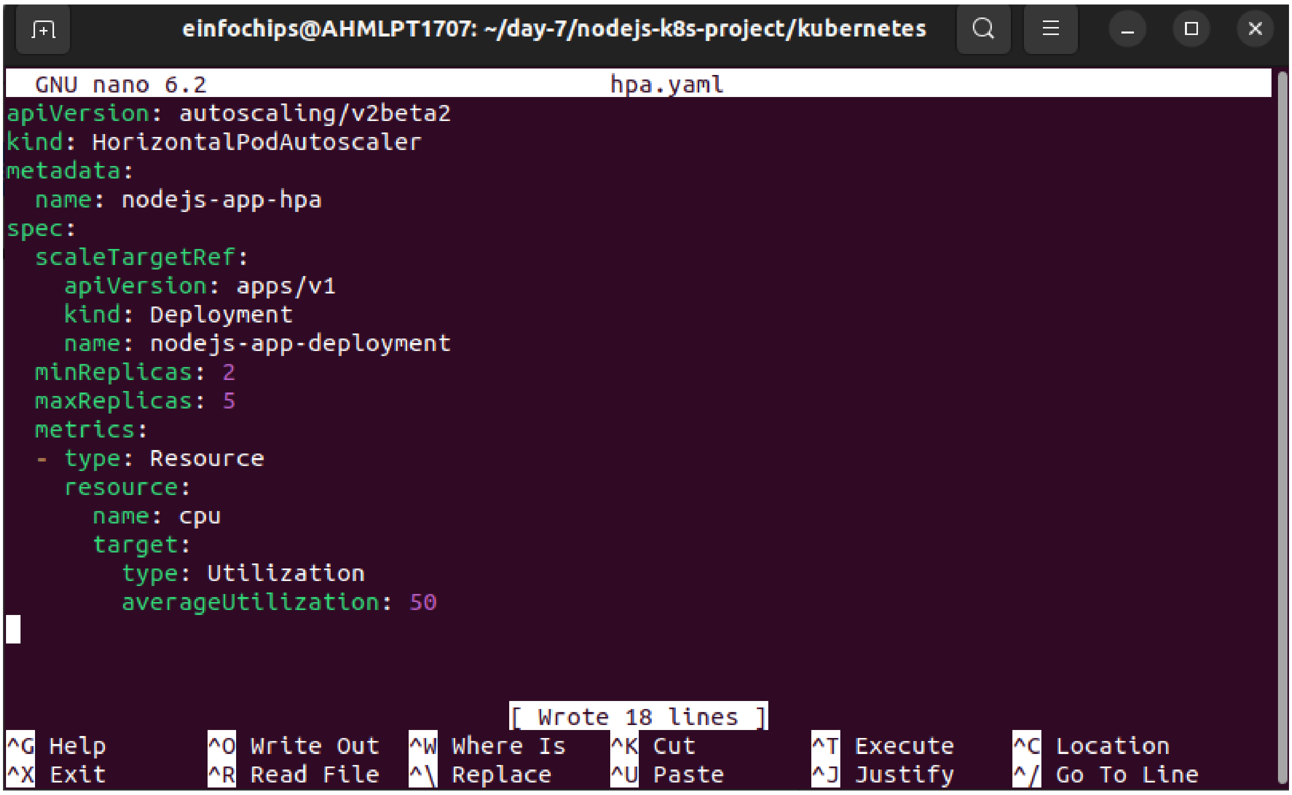


```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ minikube version
minikube version: v1.33.1
commit: 5883c09216182566a63dff4c326a6fc9ed2982ff
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04
✨  Using the docker driver based on existing profile
👍  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" namespace by
 default
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/configmap.
yaml
configmap/app-config created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/secret.yam
l
secret/app-secrets created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/deployment
.yaml
deployment.apps/nodejs-app-deployment created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

Implement Autoscaling

Create Horizontal Pod Autoscaler

Create kubernetes/hpa.yaml:



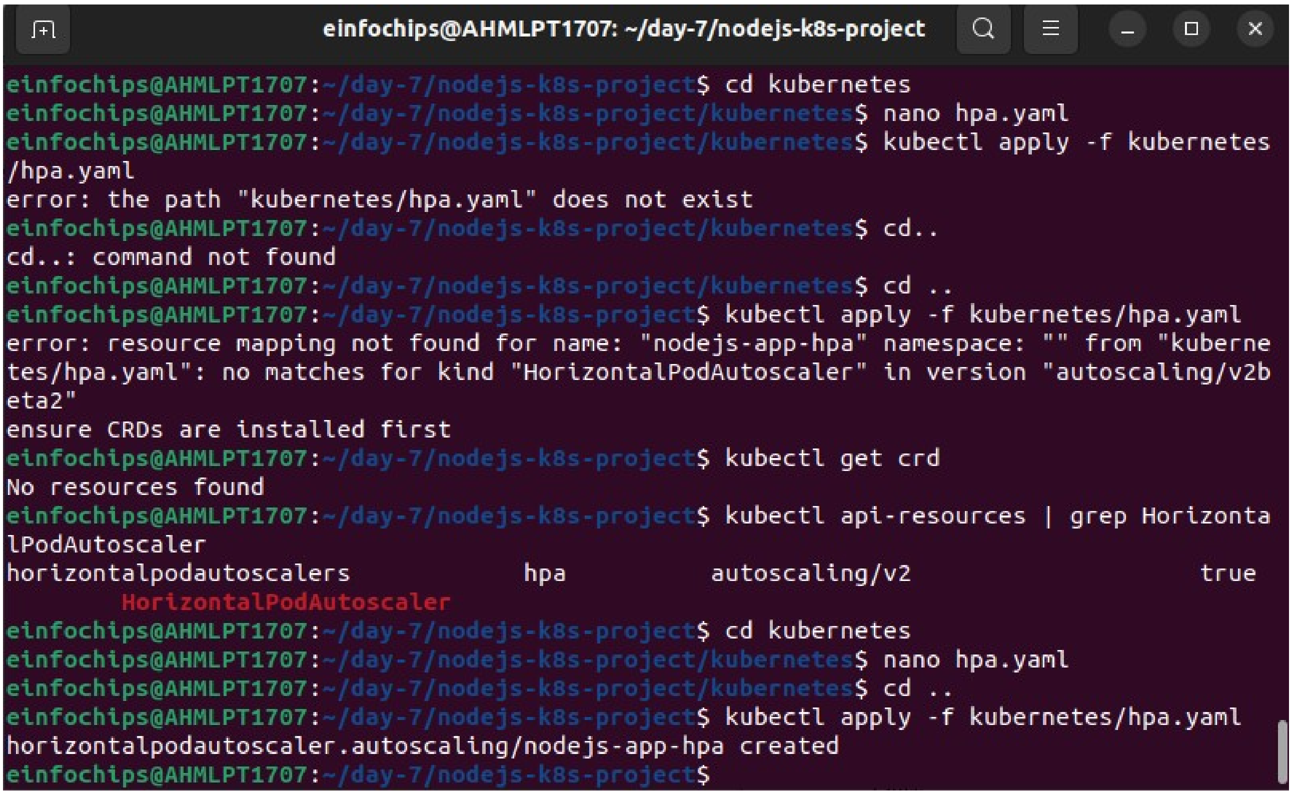
einfochips@AHMLPT1707: ~/day-7/nodejs-k8s-project/kubernetes

```
  GNU nano 6.2                          hpa.yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-app-deployment
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50


                                   [ Wrote 18 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line
```

here we need to change v2beta2 to v2

Apply the VPA:

```
kubectl apply -f kubernetes/vpa.yaml
```

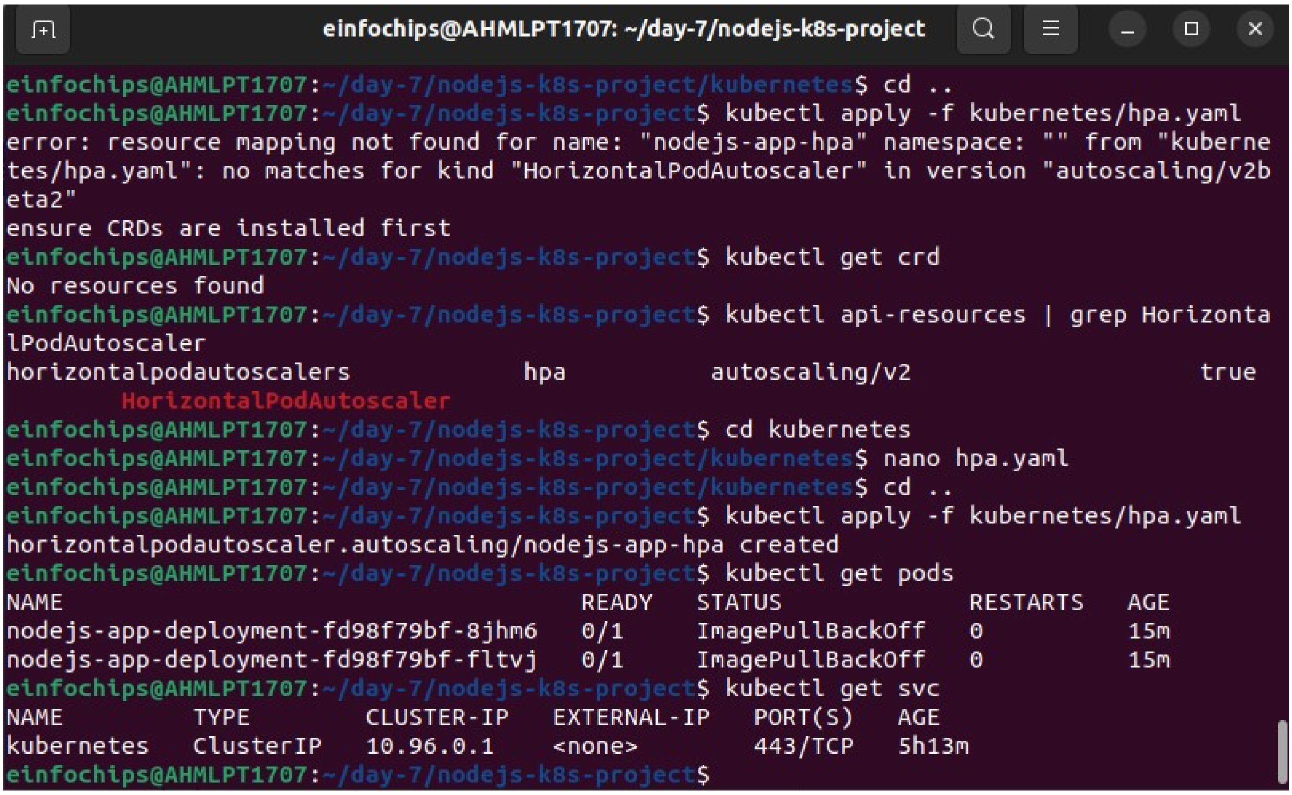


```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ kubectl apply -f kubernetes
/hpa.yaml
error: the path "kubernetes/hpa.yaml" does not exist
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd..
cd..: command not found
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
error: resource mapping not found for name: "nodejs-app-hpa" namespace: "" from "kuberne
tes/hpa.yaml": no matches for kind "HorizontalPodAutoscaler" in version "autoscaling/v2b
eta2"
ensure CRDs are installed first
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get crd
No resources found
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl api-resources | grep Horizonta
lPodAutoscaler
horizontalpodautoscalers          hpa         autoscaling/v2                    true
        HorizontalPodAutoscaler
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-app-hpa created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

Test the Deployment

7.1 Check the Status of Pods, Services, and HPA

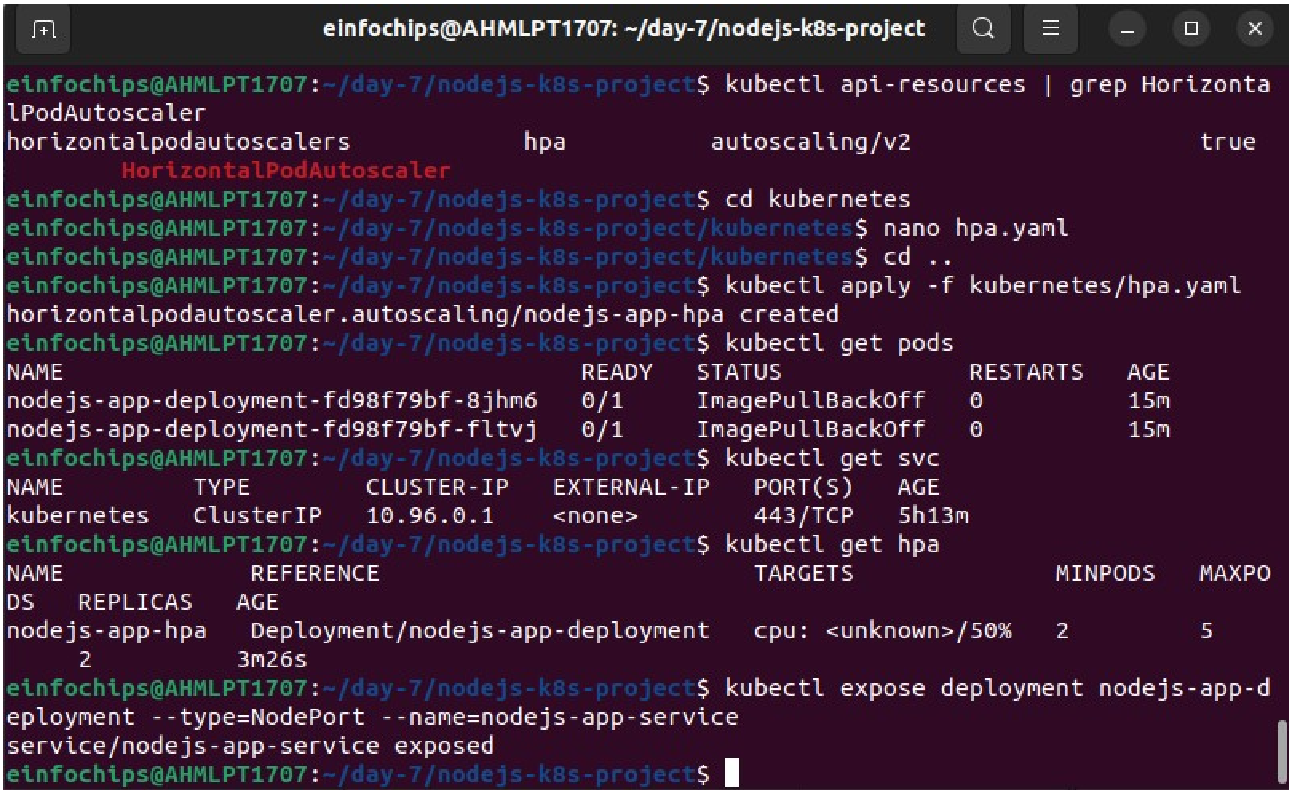
Verify the Pods and service:



```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
error: resource mapping not found for name: "nodejs-app-hpa" namespace: "" from "kuberne
tes/hpa.yaml": no matches for kind "HorizontalPodAutoscaler" in version "autoscaling/v2b
eta2"
ensure CRDs are installed first
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get crd
No resources found
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl api-resources | grep Horizonta
lPodAutoscaler
horizontalpodautoscalers          hpa         autoscaling/v2                    true
        HorizontalPodAutoscaler
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-app-hpa created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get pods
NAME                                     READY   STATUS             RESTARTS   AGE
nodejs-app-deployment-fd98f79bf-8jhm6    0/1     ImagePullBackOff   0          15m
nodejs-app-deployment-fd98f79bf-fltvj    0/1     ImagePullBackOff   0          15m
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   5h13m
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

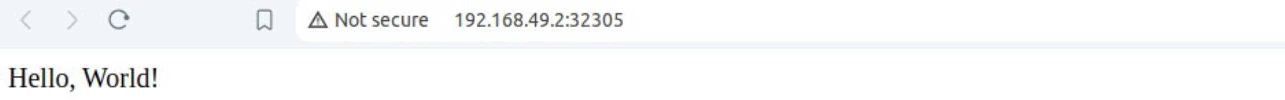
Access the Application

Expose the Service:



```
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl api-resources | grep HorizontalPodAutoscaler
horizontalpodautoscalers            hpa          autoscaling/v2                     true
        HorizontalPodAutoscaler
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ cd kubernetes
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project/kubernetes$ cd ..
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl apply -f kubernetes/hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-app-hpa created
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get pods
NAME                                        READY   STATUS             RESTARTS   AGE
nodejs-app-deployment-fd98f79bf-8jhm6       0/1     ImagePullBackOff   0          15m
nodejs-app-deployment-fd98f79bf-fltvj       0/1     ImagePullBackOff   0          15m
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   5h13m
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl get hpa
NAME             REFERENCE                          TARGETS             MINPODS   MAXPODS   REPLICAS   AGE
nodejs-app-hpa   Deployment/nodejs-app-deployment   cpu: <unknown>/50%   2         5         2          3m26s
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ kubectl expose deployment nodejs-app-deployment --type=NodePort --name=nodejs-app-service
service/nodejs-app-service exposed
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$
```

Get the Minikube IP and Service Port:



192.168.49.2:32305

Hello, World!

Project 02

Deploy a Node.js application to Kubernetes with advanced usage of ConfigMaps and Secrets. Implement Horizontal Pod Autoscaler (HPA) with both scale-up and scale-down policies. The project will include a multi-environment configuration strategy, integrating a Redis cache, and monitoring application metrics.

Project Setup

1.1 Initialize a Git Repository

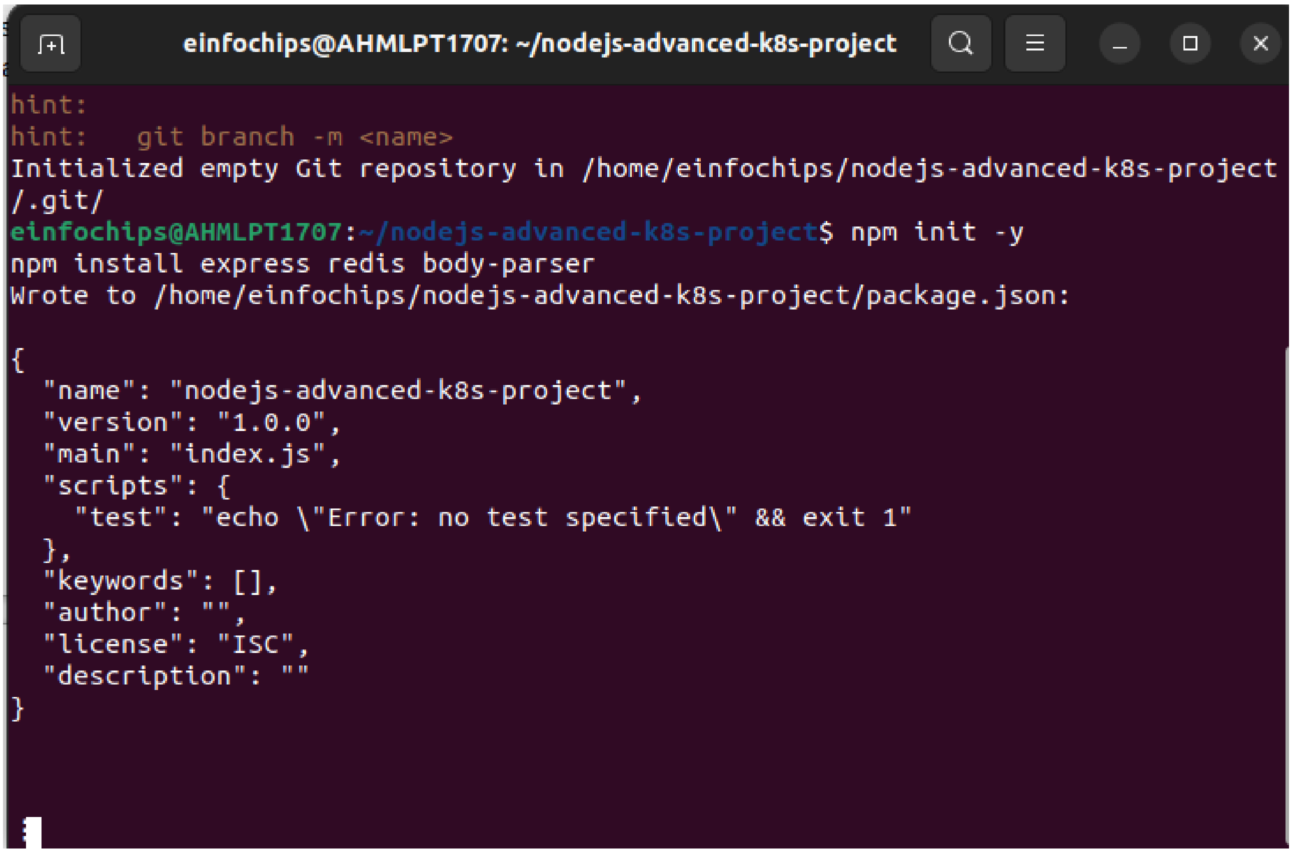
Create a new directory for your project and initialize Git:

```
mkdir nodejs-advanced-k8s-project
cd nodejs-advanced-k8s-project
git init
```

1.2 Create Initial Files

Create the initial Node.js application and Docker-related files:

```
npm init -y
npm install express redis body-parser
```



```
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/einfochips/nodejs-advanced-k8s-project
/.git/
einfochips@AHMLPT1707:~/nodejs-advanced-k8s-project$ npm init -y
npm install express redis body-parser
Wrote to /home/einfochips/nodejs-advanced-k8s-project/package.json:

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

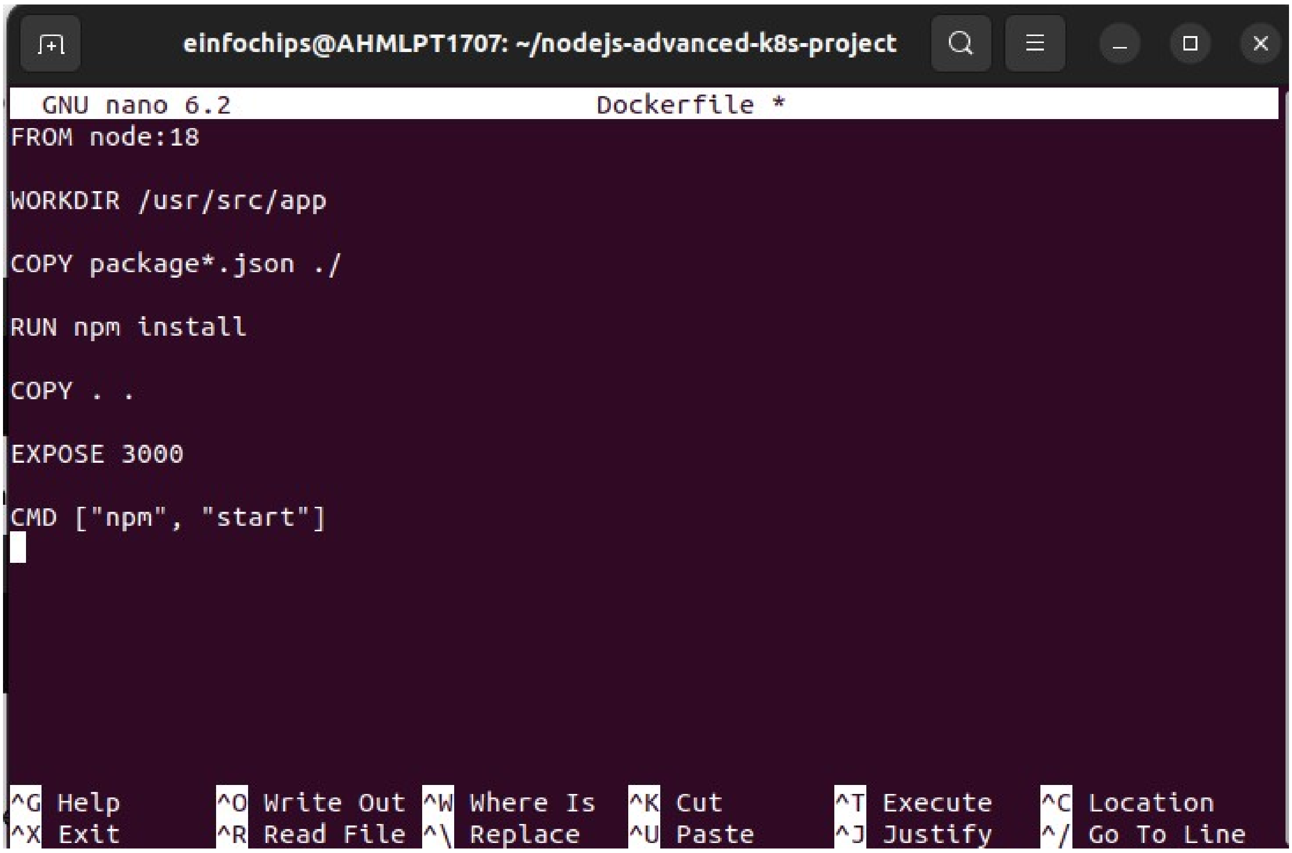
create app.js

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

const bodyParser = require('body-parser');
const redis = require('redis');
const app = express();
const PORT = process.env.PORT || 3000;

// Connect to Redis
```

```
const redisClient = redis.createClient({
  url: `redis://${process.env.REDIS_HOST}:${process.env.REDIS_PORT}`
});
redisClient.on('error', (err) => console.error('Redis Client Error', err));

app.use(bodyParser.json());

app.get('/', async (req, res) => {
  const visits = await redisClient.get('visits');
  if (visits) {
    await redisClient.set('visits', parseInt(visits) + 1);
  } else {
    await redisClient.set('visits', 1);
  }
  res.send(`Hello, World! You are visitor number ${visits || 1}`);
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

create dockerfile

```
FROM node:18

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
```



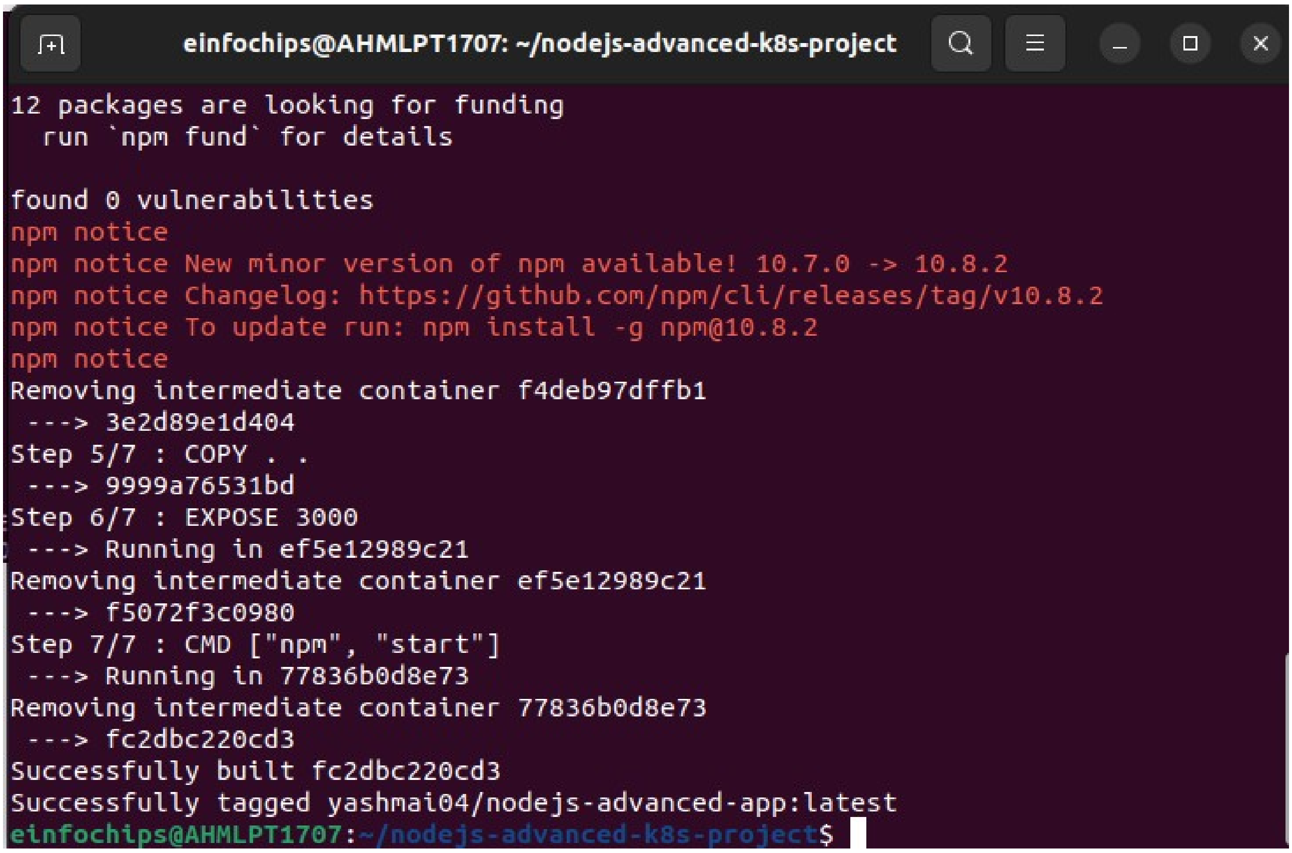
```
  GNU nano 6.2                    Dockerfile *
FROM node:18

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
```

create .dockerignore

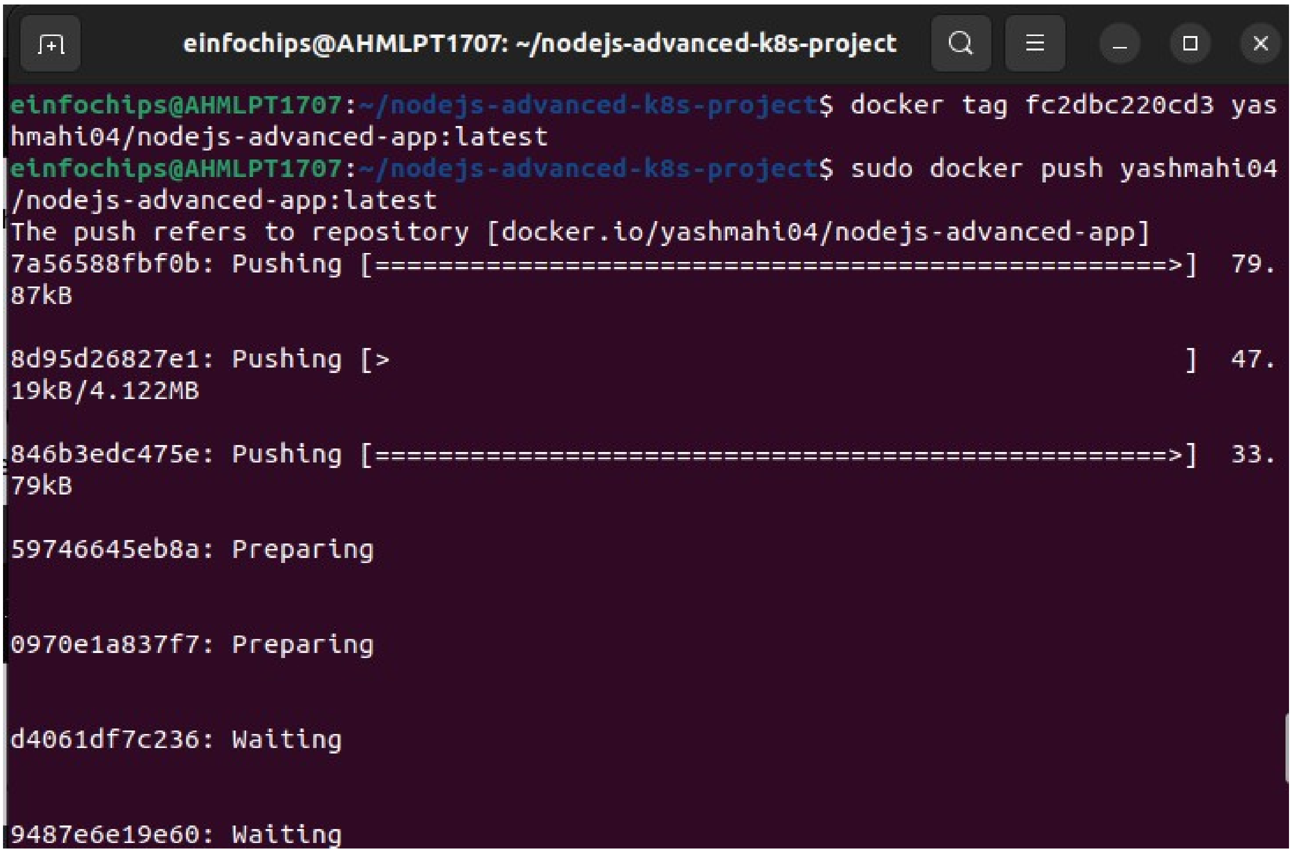
```
node_modules
.npm
```

1. Build and push Docker image:

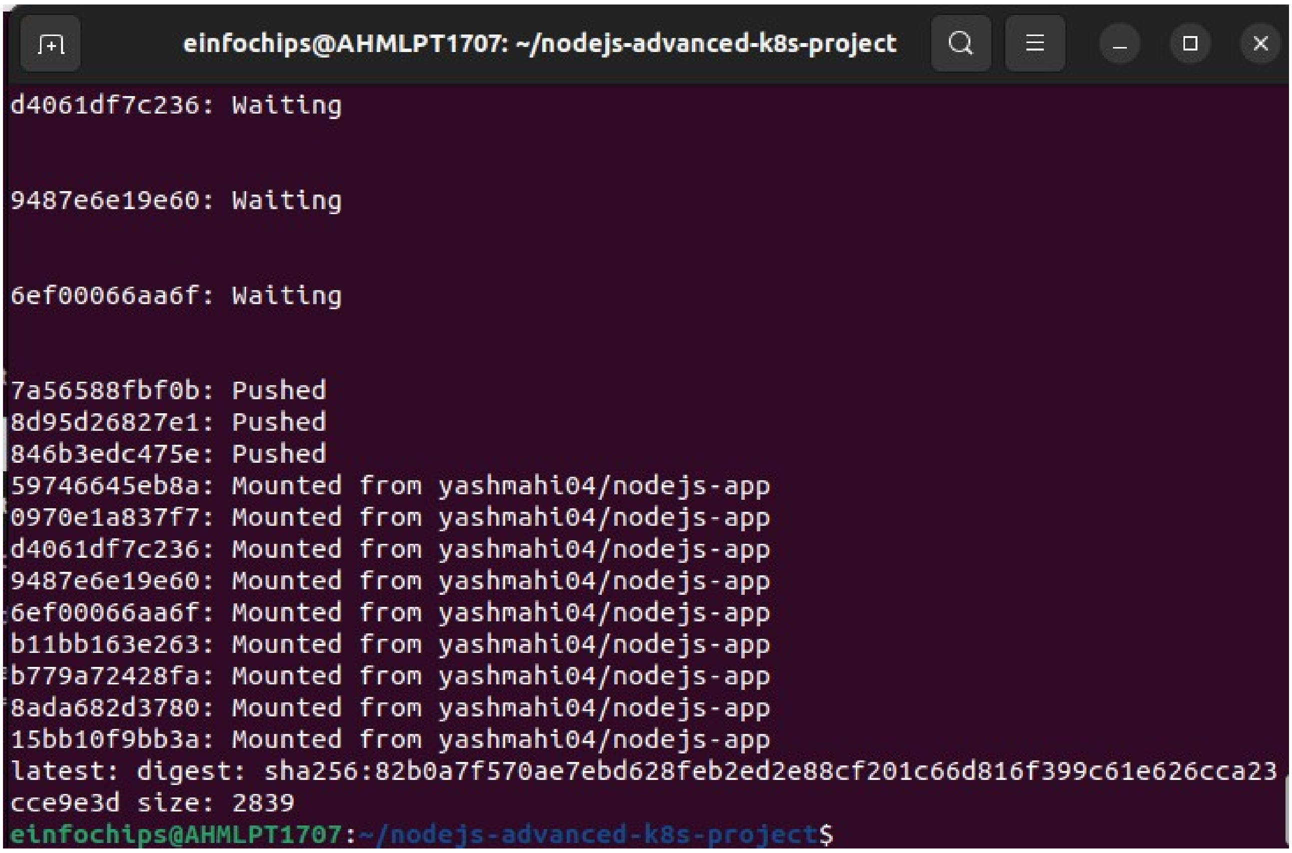
```
docker build -t yashmai04/nodejs-advanced-app:latest .
docker push yashmahi04/nodejs-advanced-app:latest
```



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

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice
Removing intermediate container f4deb97dffb1
 ---> 3e2d89e1d404
Step 5/7 : COPY . .
 ---> 9999a76531bd
Step 6/7 : EXPOSE 3000
 ---> Running in ef5e12989c21
Removing intermediate container ef5e12989c21
 ---> f5072f3c0980
Step 7/7 : CMD ["npm", "start"]
 ---> Running in 77836b0d8e73
Removing intermediate container 77836b0d8e73
 ---> fc2dbc220cd3
Successfully built fc2dbc220cd3
Successfully tagged yashmai04/nodejs-advanced-app:latest
einfochips@AHMLPT1707:~/nodejs-advanced-k8s-project$
```



```
einfochips@AHMLPT1707:~/nodejs-advanced-k8s-project$ docker tag fc2dbc220cd3 yas
hmahi04/nodejs-advanced-app:latest
einfochips@AHMLPT1707:~/nodejs-advanced-k8s-project$ sudo docker push yashmahi04
/nodejs-advanced-app:latest
The push refers to repository [docker.io/yashmahi04/nodejs-advanced-app]
7a56588fbf0b: Pushing [==================================================>]  79.
87kB

8d95d26827e1: Pushing [>                                                  ]  47.
19kB/4.122MB

846b3edc475e: Pushing [==================================================>]  33.
79kB

59746645eb8a: Preparing


0970e1a837f7: Preparing


d4061df7c236: Waiting


9487e6e19e60: Waiting
```



```
d4061df7c236: Waiting


9487e6e19e60: Waiting


6ef00066aa6f: Waiting


7a56588fbf0b: Pushed
8d95d26827e1: Pushed
846b3edc475e: Pushed
59746645eb8a: Mounted from yashmahi04/nodejs-app
0970e1a837f7: Mounted from yashmahi04/nodejs-app
d4061df7c236: Mounted from yashmahi04/nodejs-app
9487e6e19e60: Mounted from yashmahi04/nodejs-app
6ef00066aa6f: Mounted from yashmahi04/nodejs-app
b11bb163e263: Mounted from yashmahi04/nodejs-app
b779a72428fa: Mounted from yashmahi04/nodejs-app
8ada682d3780: Mounted from yashmahi04/nodejs-app
15bb10f9bb3a: Mounted from yashmahi04/nodejs-app
latest: digest: sha256:82b0a7f570ae7ebd628feb2ed2e88cf201c66d816f399c61e626cca23
cce9e3d size: 2839
einfochips@AHMLPT1707:~/nodejs-advanced-k8s-project$
```

2. Advanced Kubernetes Configuration

2.1 Deployment Configuration

Create `kubernetes/deployment.yaml` to deploy the Node.js application with Redis dependency:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-advanced-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-advanced-app
  template:
    metadata:
      labels:
        app: nodejs-advanced-app
    spec:
      containers:
      - name: nodejs-advanced-app
        image: your-dockerhub-username/nodejs-advanced-app:latest
        ports:
        - containerPort: 3000
        env:
        - name: PORT
         valueFrom:
          configMapKeyRef:
           name: app-config
           key: PORT
        - name: REDIS_HOST
         valueFrom:
          configMapKeyRef:
           name: redis-config
           key: REDIS_HOST
        - name: REDIS_PORT
         valueFrom:
          configMapKeyRef:
           name: redis-config
           key: REDIS_PORT
        - name: NODE_ENV
         valueFrom:
          secretKeyRef:
           name: app-secrets
           key: NODE_ENV
      - name: redis
        image: redis:latest
        ports:
        - containerPort: 6379
```

2.2 ConfigMap for Application and Redis

Create `kubernetes/configmap.yaml` to manage application and Redis configurations:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: redis-config
data:
  REDIS_HOST: "redis"
  REDIS_PORT: "6379"
```

2.3 Secret for Sensitive Data

Create kubernetes/secret.yaml to manage sensitive environment variables:

```
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbg== # Base64 encoded value for "production"
```

2.4 Service Configuration

Create kubernetes/service.yaml to expose the Node.js application:

```
apiVersion: v1
kind: Service
metadata:
  name: nodejs-advanced-app-service
```

```
spec:
  selector:
    app: nodejs-advanced-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 3000
  type: LoadBalancer
```

2.5 Horizontal Pod Autoscaler with Scale-Up and Scale-Down Policies

Create kubernetes/hpa.yaml to manage autoscaling:

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-advanced-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-advanced-app-deployment
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
  - type: Resource
    resource:
      name: memory
      target:
        type: Utilization
        averageUtilization: 70
  behavior:
    scaleUp:
      stabilizationWindowSeconds: 30
      selectPolicy: Max
      policies:
      - type: Pods
        value: 2
        periodSeconds: 30
      - type: Resource
        resource: cpu
        value: 2
        periodSeconds: 30
    scaleDown:
      stabilizationWindowSeconds: 30
      selectPolicy: Min
      policies:
      - type: Pods
        value: 1
        periodSeconds: 30
      - type: Resource
        resource: memory
        value: 1
        periodSeconds: 30
```

2.6 Vertical Pod Autoscaler Configuration

Create `kubernetes/vpa.yaml` to manage vertical scaling:

```
apiVersion: autoscaling.k8s.io/v1beta2
kind: VerticalPodAutoscaler
metadata:
  name: nodejs-advanced-app-vpa
spec:
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-advanced-app-deployment
  updatePolicy:
    updateMode: "Auto"
```

2.7 Redis Deployment

Add a Redis deployment configuration to kubernetes/redis-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: redis
        image: redis:latest
        ports:
        - containerPort: 6379
```

Add Redis service configuration to kubernetes/redis-service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: redis-service
spec:
  selector:
    app: redis
  ports:
  - protocol: TCP
    port: 6379
    targetPort: 6379
  type: ClusterIP
```

2.8 Apply Kubernetes Configurations

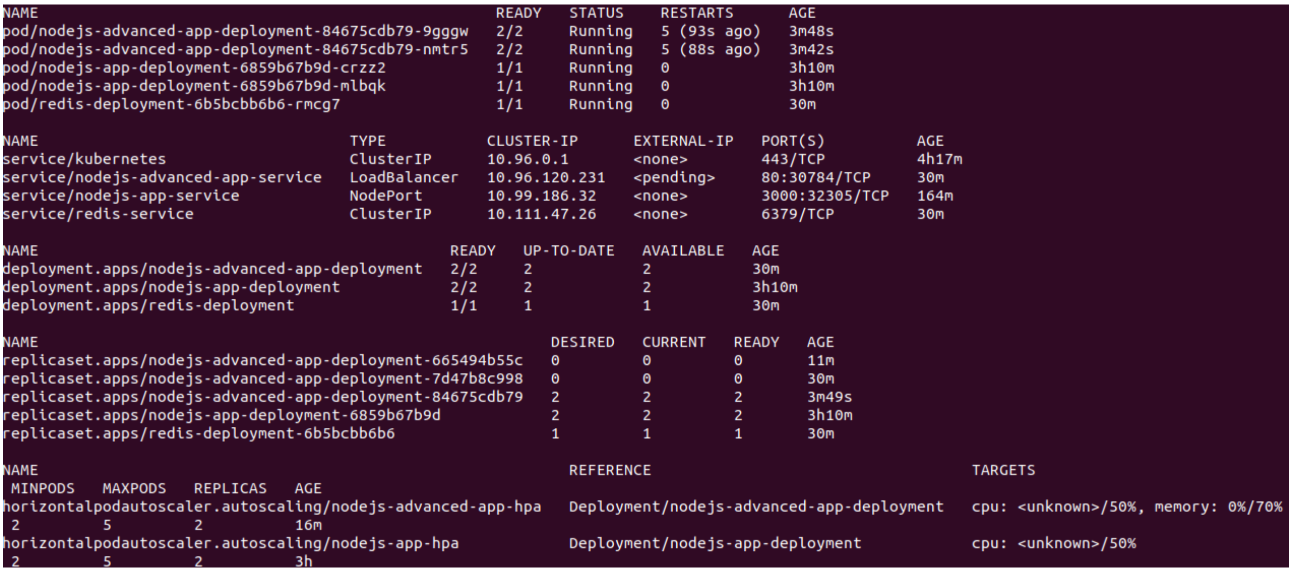
- Apply all configurations to your Minikube cluster:

```
kubectl apply -f kubernetes/redis-deployment.yaml
kubectl apply -f kubernetes/redis-service.yaml
kubectl apply -f kubernetes/configmap.yaml
kubectl apply -f kubernetes/secret.yaml
kubectl apply -f kubernetes/deployment.yaml
kubectl apply -f kubernetes/service.yaml
kubectl apply -f kubernetes/hpa.yaml
kubectl apply -f kubernetes/vpa.yaml
```

2.9 Verify Deployments and Services

- Check the status of your deployments and services:

```
kubectl get all
```



```
NAME                                                READY   STATUS    RESTARTS      AGE
pod/nodejs-advanced-app-deployment-84675cdb79-9gggw   2/2     Running   5 (93s ago)   3m48s
pod/nodejs-advanced-app-deployment-84675cdb79-nmtr5   2/2     Running   5 (88s ago)   3m42s
pod/nodejs-app-deployment-6859b67b9d-crzz2            1/1     Running   0             3h10m
pod/nodejs-app-deployment-6859b67b9d-mlbqk            1/1     Running   0             3h10m
pod/redis-deployment-6b5bcbb6b6-rmcg7                 1/1     Running   0             30m

NAME                                  TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                    ClusterIP      10.96.0.1        <none>        443/TCP          4h17m
service/nodejs-advanced-app-service   LoadBalancer   10.96.120.231    <pending>     80:30784/TCP     30m
service/nodejs-app-service            NodePort       10.99.186.32     <none>        3000:32305/TCP   164m
service/redis-service                 ClusterIP      10.111.47.26     <none>        6379/TCP         30m

NAME                                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-advanced-app-deployment   2/2     2            2           30m
deployment.apps/nodejs-app-deployment            2/2     2            2           3h10m
deployment.apps/redis-deployment                 1/1     1            1           30m

NAME                                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/nodejs-advanced-app-deployment-665494b55c   0         0         0       11m
replicaset.apps/nodejs-advanced-app-deployment-7d47b8c998   0         0         0       30m
replicaset.apps/nodejs-advanced-app-deployment-84675cdb79   2         2         2       3m49s
replicaset.apps/nodejs-app-deployment-6859b67b9d            2         2         2       3h10m
replicaset.apps/redis-deployment-6b5bcbb6b6                 1         1         1       30m

NAME                                                          REFERENCE                                   TARGETS
 MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/nodejs-advanced-app-hpa   Deployment/nodejs-advanced-app-deployment   cpu: <unknown>/50%, memory: 0%/70%
 2         5         2          16m
horizontalpodautoscaler.autoscaling/nodejs-app-hpa            Deployment/nodejs-app-deployment            cpu: <unknown>/50%
 2         5         2          3h
```

Access the application via Minikube:

```
minikube service nodejs-advanced-app-service --url
```

2.10 Testing Scaling

- Simulate load on the application to test the HPA:

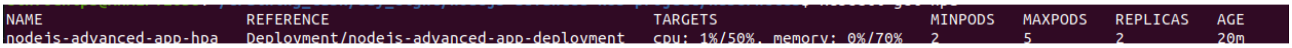
```
kubectl run -i --tty --rm load-generator --image=busybox --restart=Never --
/bin/sh
```

```
# Inside the pod, run the following command to generate load
while true; do wget -q -O- http://nodejs-advanced-app-service; done
```

2.11 Validate Autoscaling Behavior

- Observe the HPA behavior:

```
kubectl get hpa
```



```
NAME                    REFERENCE                                  TARGETS                       MINPODS   MAXPODS   REPLICAS   AGE
nodejs-advanced-app-hpa   Deployment/nodejs-advanced-app-deployment   cpu: 1%/50%, memory: 0%/70%   2         5         2          20m
```

3. Project Wrap-Up

3.1 Review and Clean Up

- After completing the project, review the configurations and clean up the Minikube environment if needed:

```
minikube delete
```