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

```
  GNU nano 6.2                          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 ]
^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
```

Update package.json

```
  GNU nano 6.2                          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"
  }
}


^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
```

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" ]

^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
```

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

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

















^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 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$
```
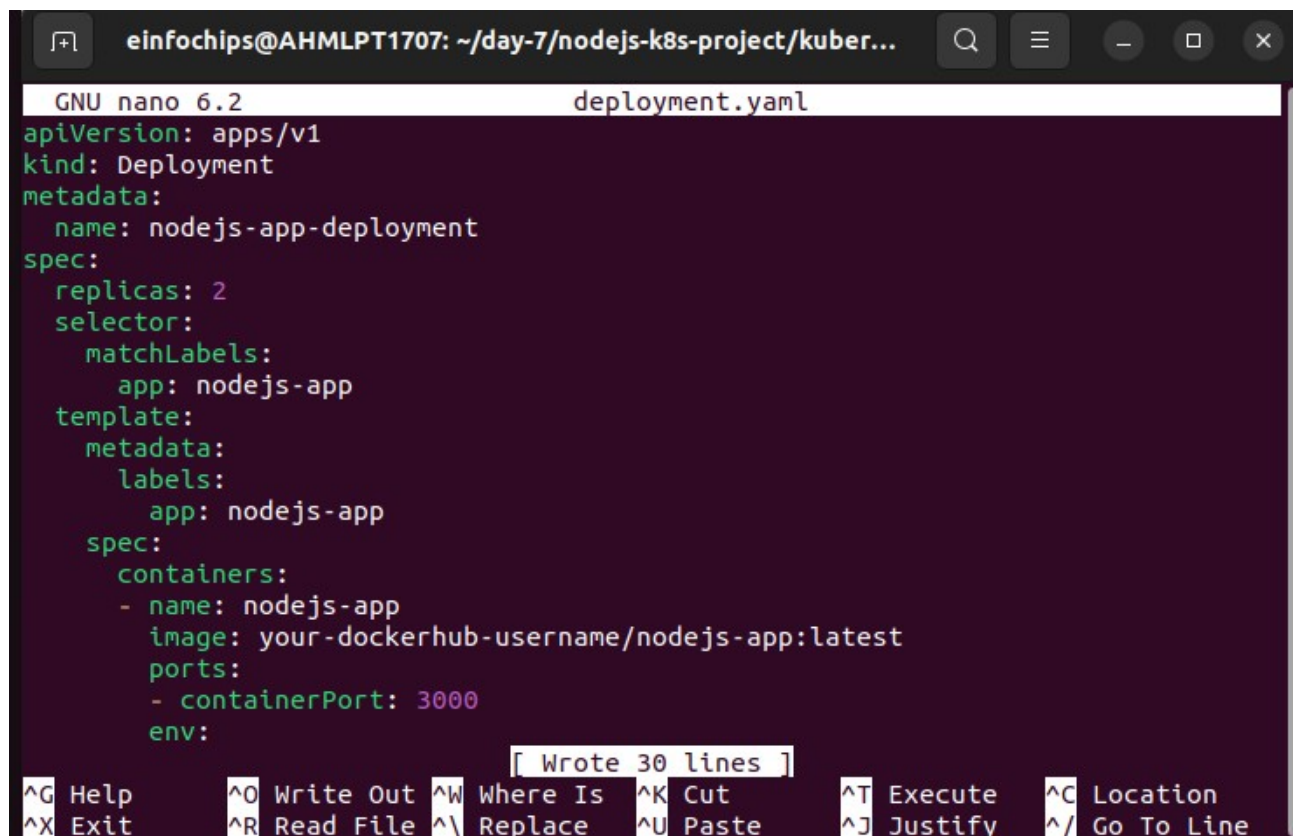
Build and Push Docker Image
 FACED ISSUE IN THIS AND THE PROCESS IS STILL STUCK HERE


```
einfochips@AHMLPT1707: ~/day-7/nodejs-k8s-project
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/2024-07-1
7T07_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


```
ushmahi04/nodejs-app:tutest
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:
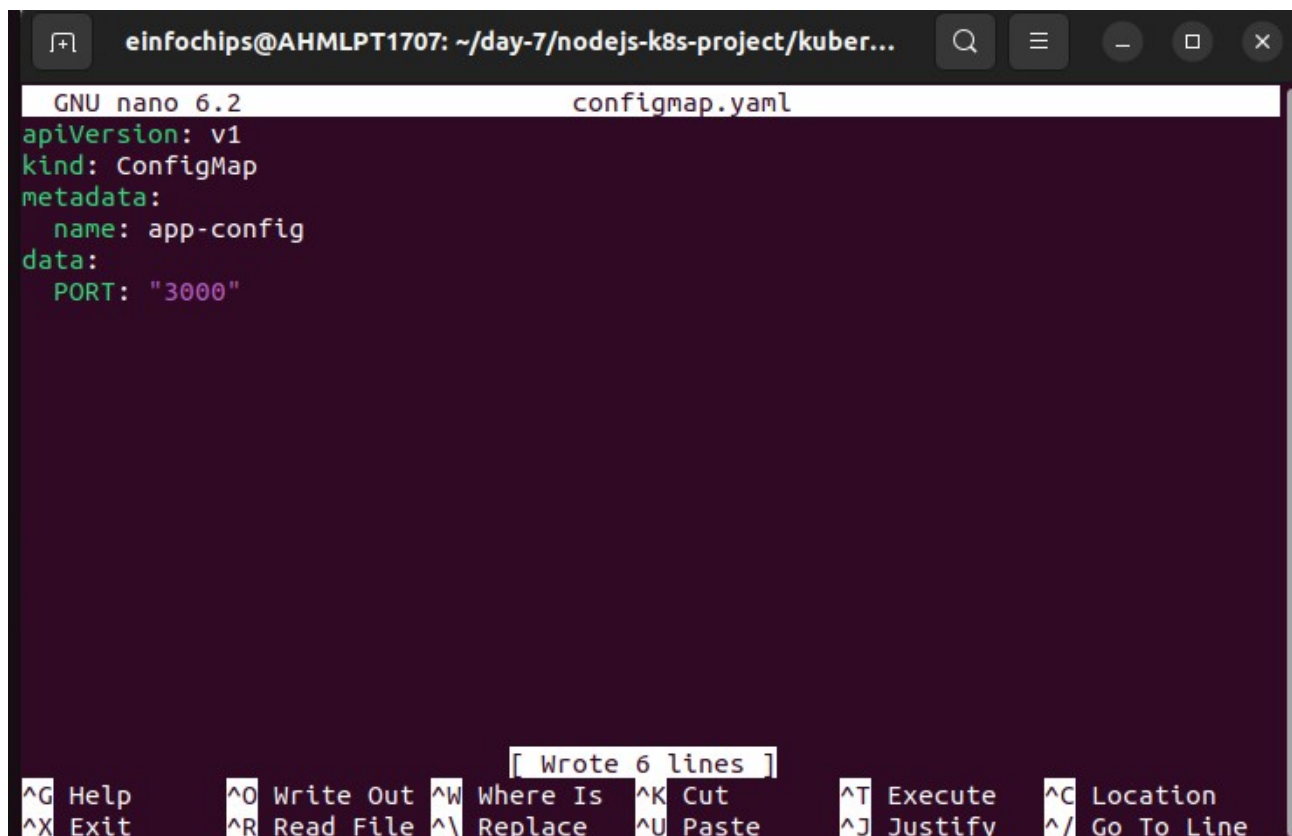
```yaml
GNU nano 6.2                          deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - 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

```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:
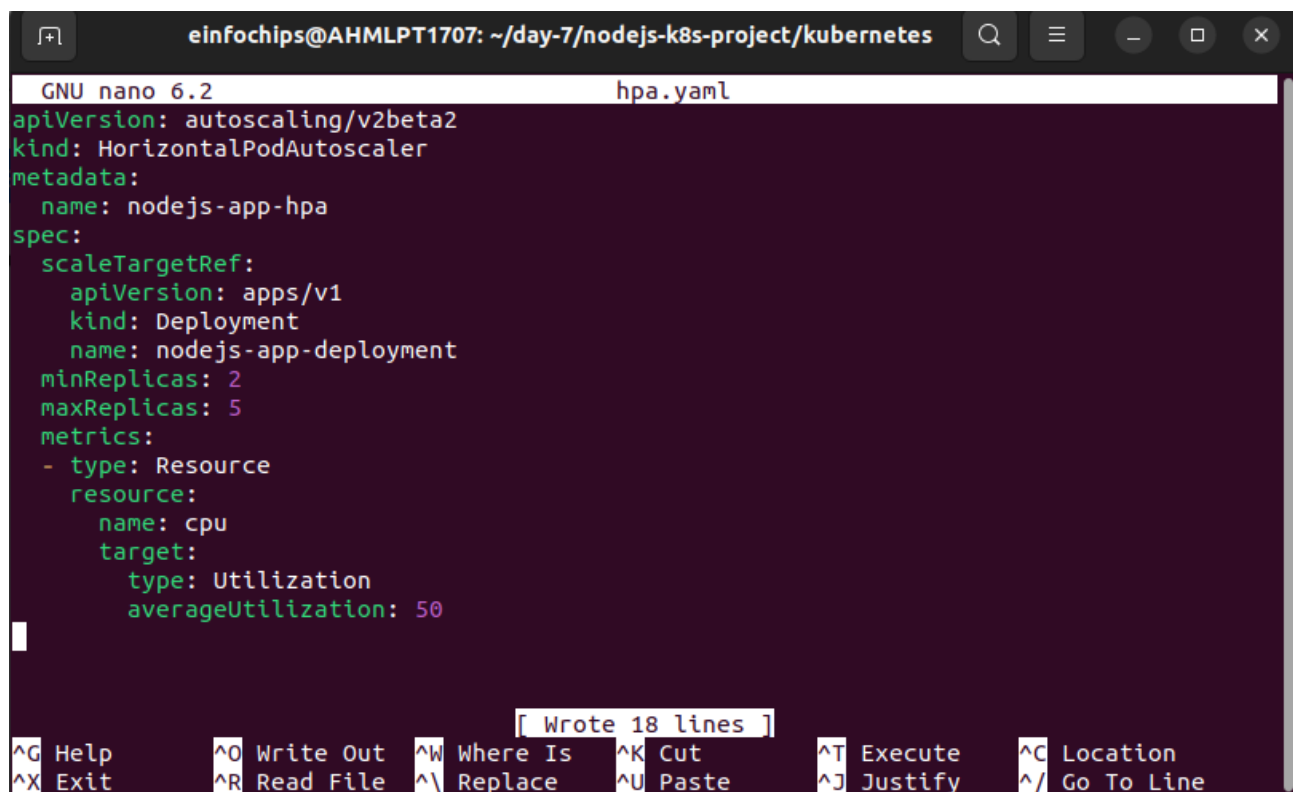

```
minkube: command not found
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ sudo apt install minikube
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package minikube
einfochips@AHMLPT1707:~/day-7/nodejs-k8s-project$ minikube version
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:

```
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

Test the Deployment

7.1 Check the Status of Pods, Services, and HPA

Verify the Pods and service:

Access the Application

Expose the Service:

Get the Minikube IP and Service Port: