Project 01

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.

Project 01

Project Steps

1. Setup Minikube and Git Repository

Start Minikube:

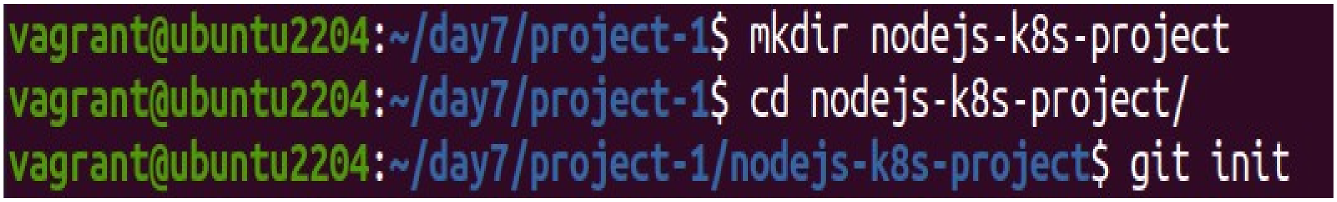
1.2 Set Up Git Repository

Create a new directory for your project:

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

Initialize Git repository:

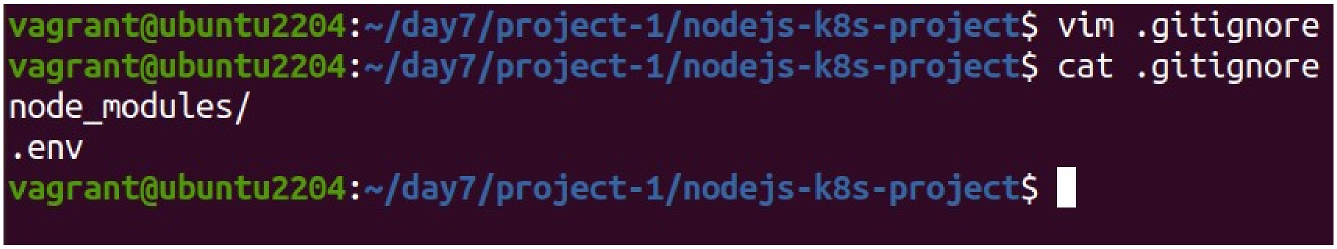
```
git init
```



```
vagrant@ubuntu2204:~/day7/project-1$ mkdir nodejs-k8s-project
vagrant@ubuntu2204:~/day7/project-1$ cd nodejs-k8s-project/
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git init
```

Create a .gitignore file:

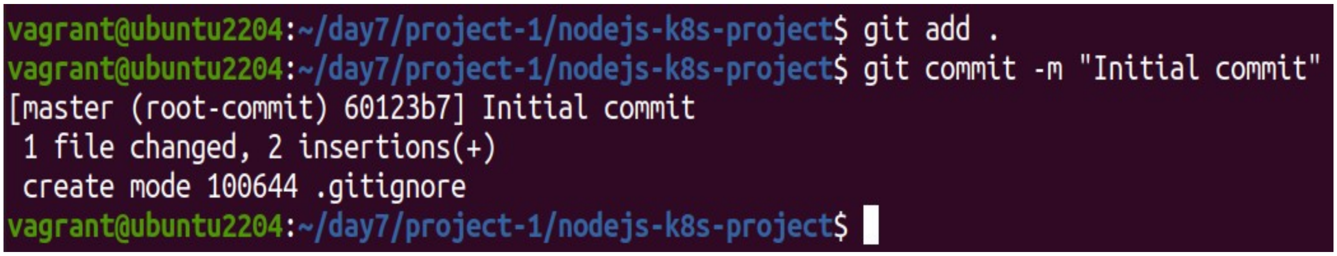
```
node_modules/
.env
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ vim .gitignore
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat .gitignore
node_modules/
.env
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

Add and commit initial changes:

```
git add .
git commit -m "Initial commit"
```



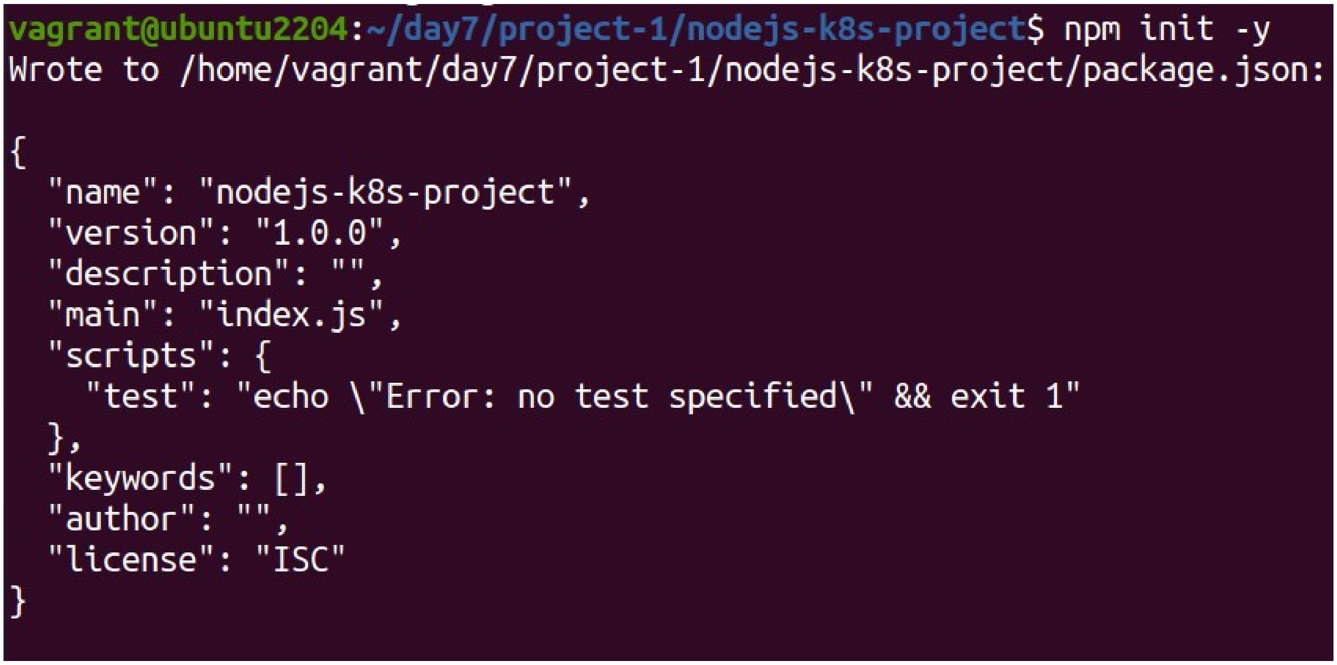
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git add .
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git commit -m "Initial commit"
[master (root-commit) 60123b7] Initial commit
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

2. Develop a Node.js Application

2.1 Create the Node.js App

Initialize the Node.js project:

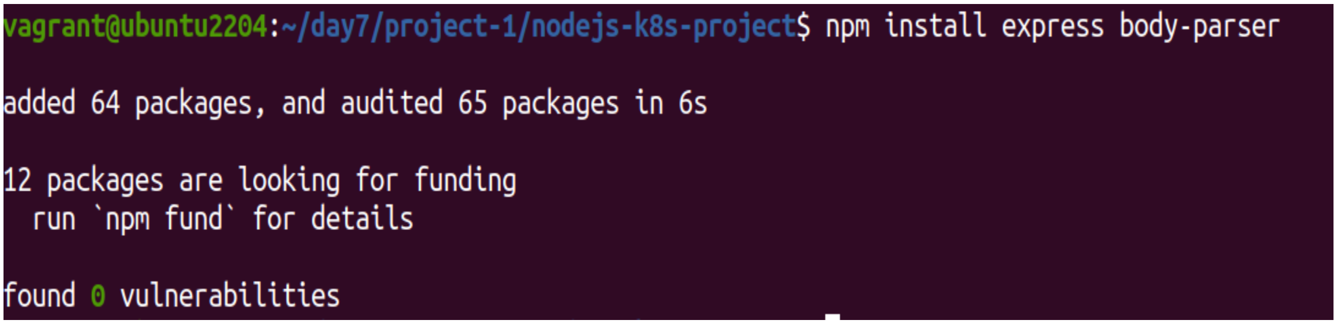
```
npm init -y
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ npm init -y
Wrote to /home/vagrant/day7/project-1/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 necessary packages:

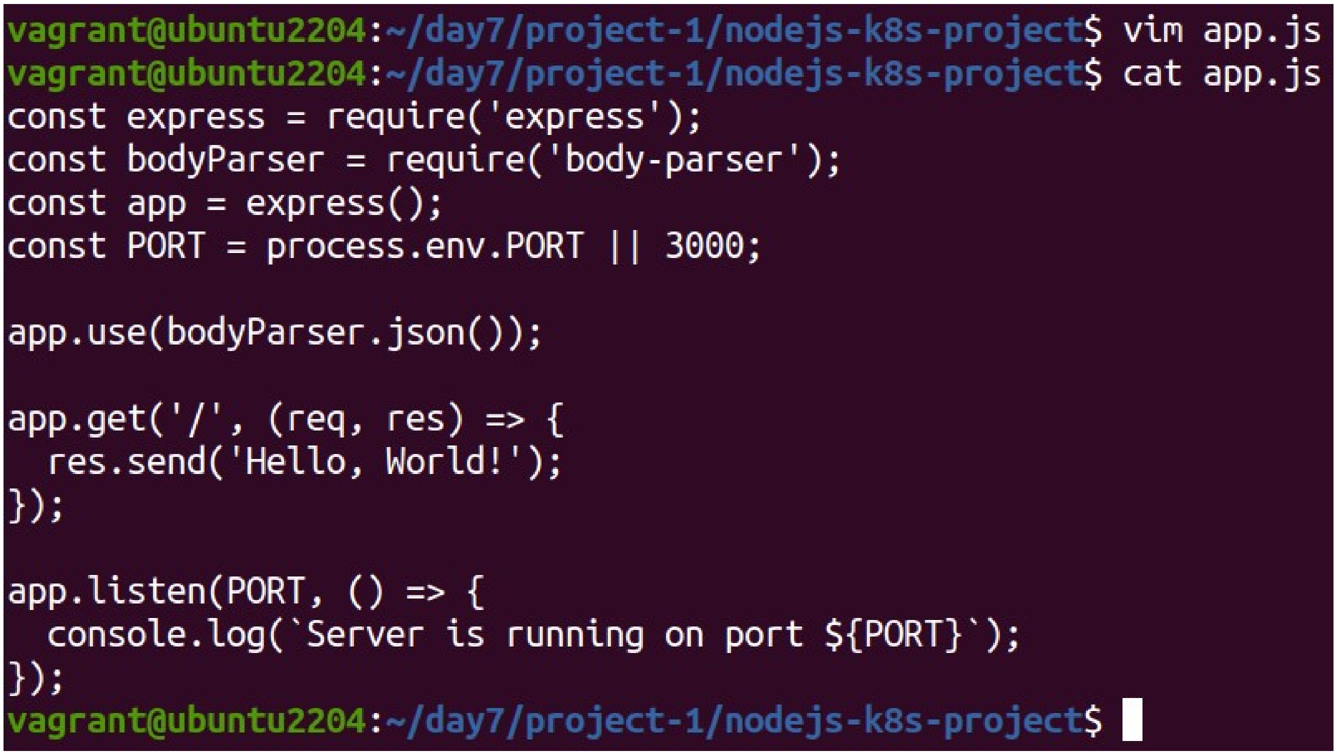
```
npm install express body-parser
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ npm install express body-parser

added 64 packages, and audited 65 packages in 6s

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

found 0 vulnerabilities
```

Create app.js:

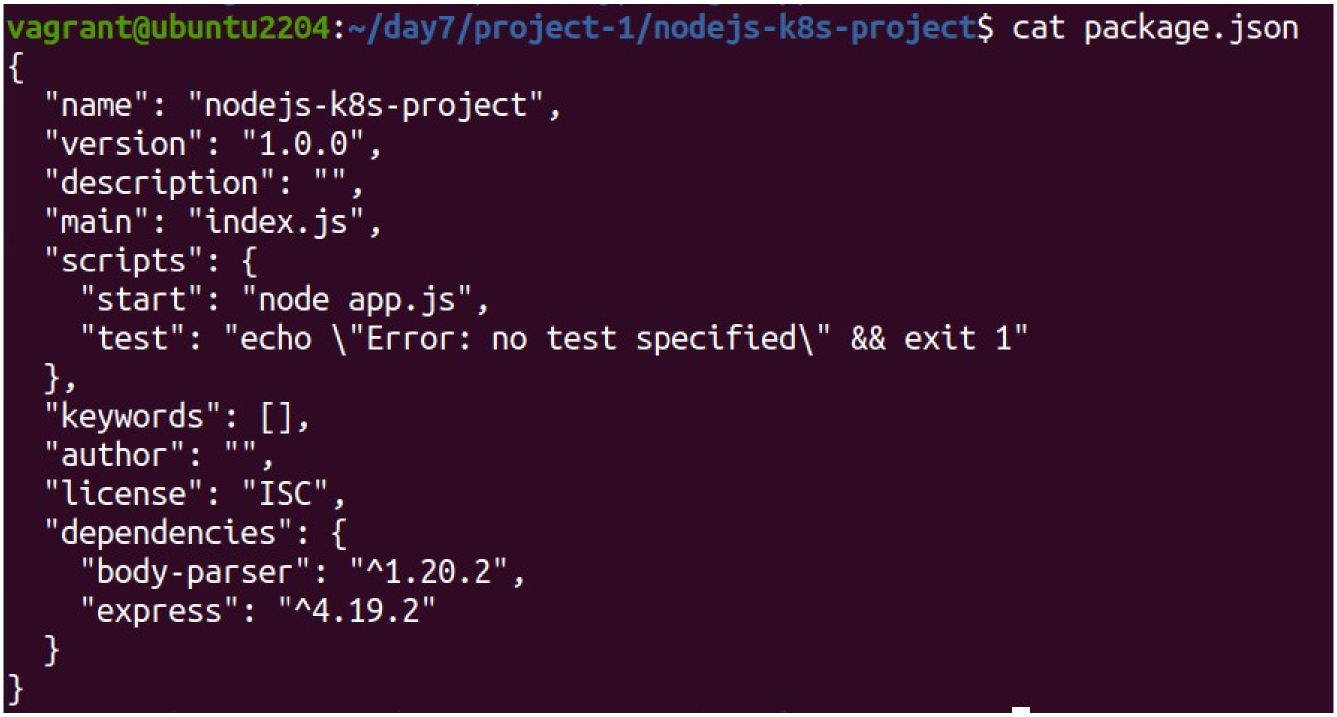
```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const PORT = process.env.PORT || 3000;


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

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



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ vim app.js
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat 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 ${PORT}`);
});
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ 
```

Update package.json to include a start script:

```
"scripts": {
  "start": "node app.js",
}
```

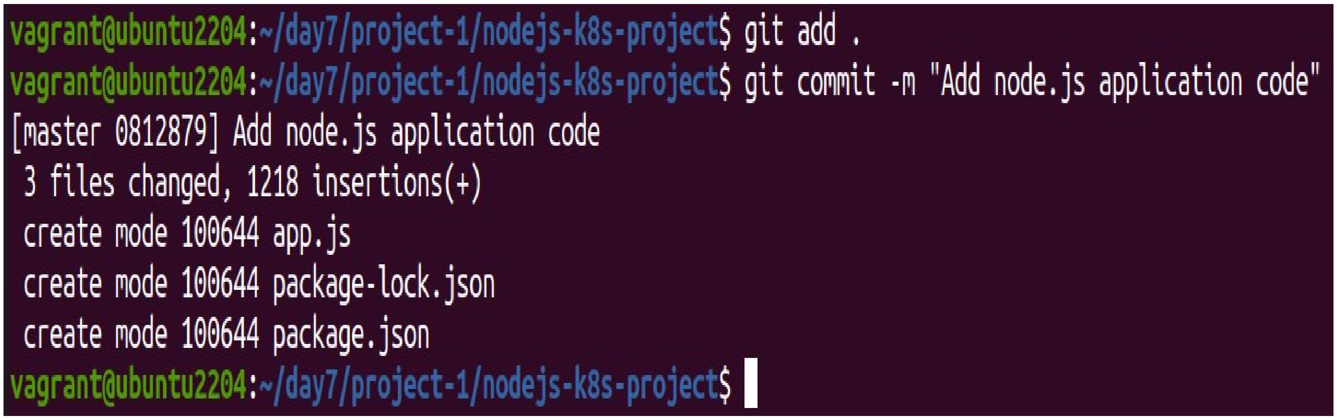


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat package.json
{
  "name": "nodejs-k8s-project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node app.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "express": "^4.19.2"
  }
}
```

2.2 Commit the Node.js Application

Add and commit changes:

```
git add .
git commit -m "Add Node.js application code"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git add .
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git commit -m "Add node.js application code"
[master 0812879] 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
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

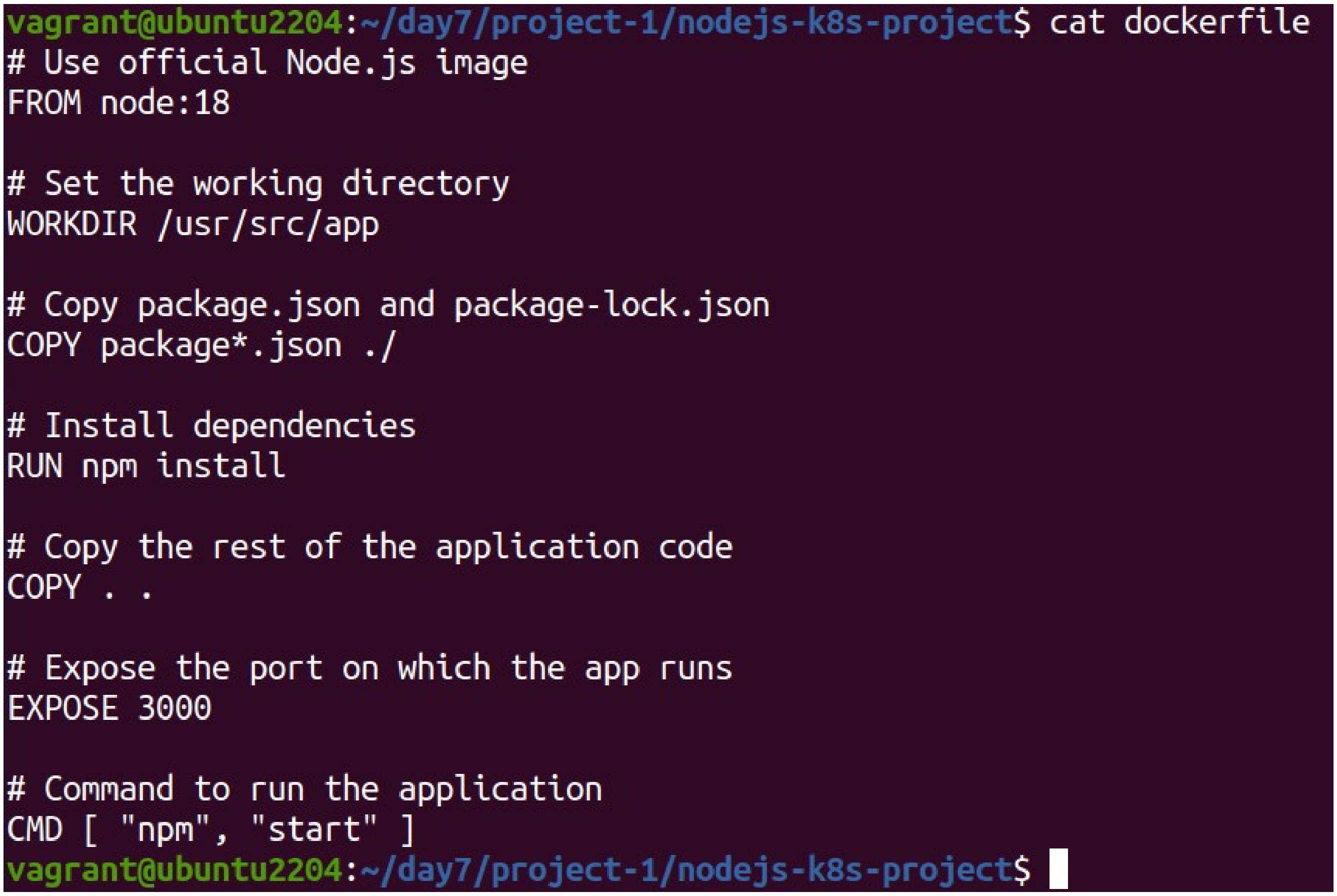
3. Create Dockerfile and Docker Compose

3.1 Create a **Dockerfile**

Add Dockerfile:

```
# Use official Node.js image
```

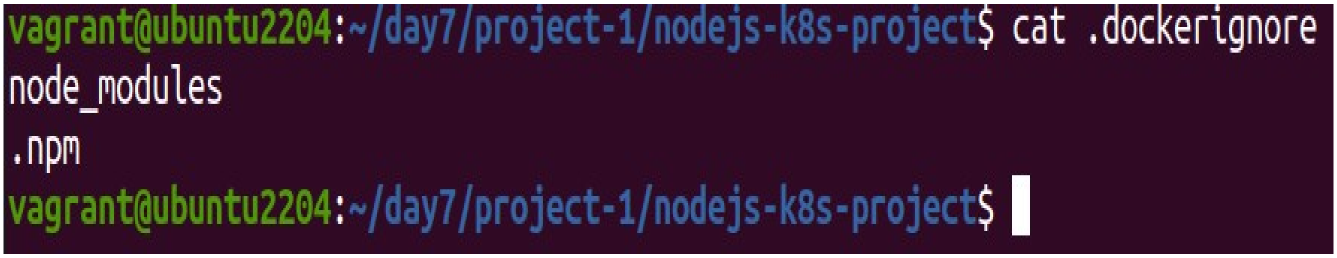
```
FROM node:18

# 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" ]
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat 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" ]
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

Create a .dockerignore file:

```
node_modules
.npm
```

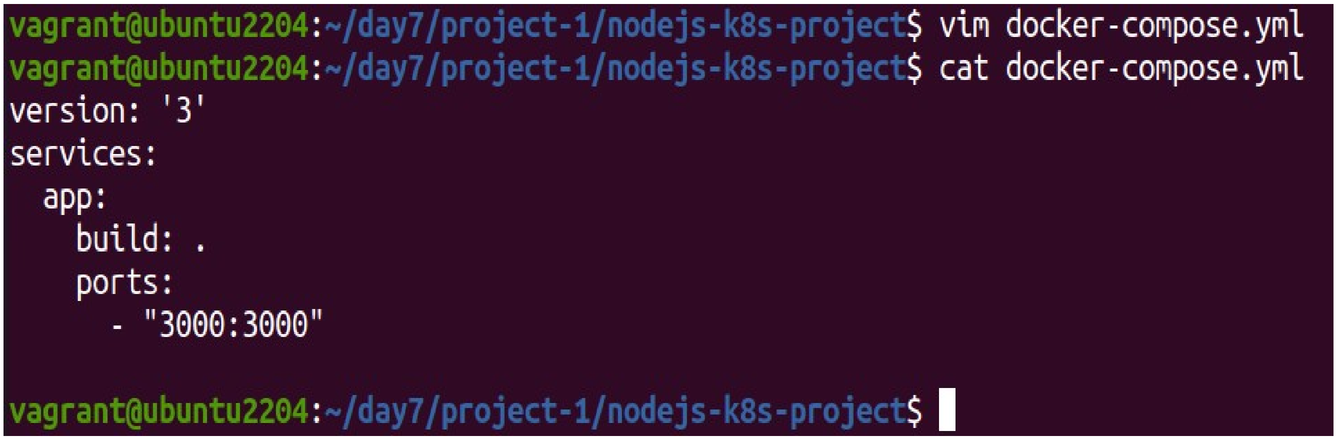


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat .dockerignore
node_modules
.npm
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

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

Add docker-compose.yml:

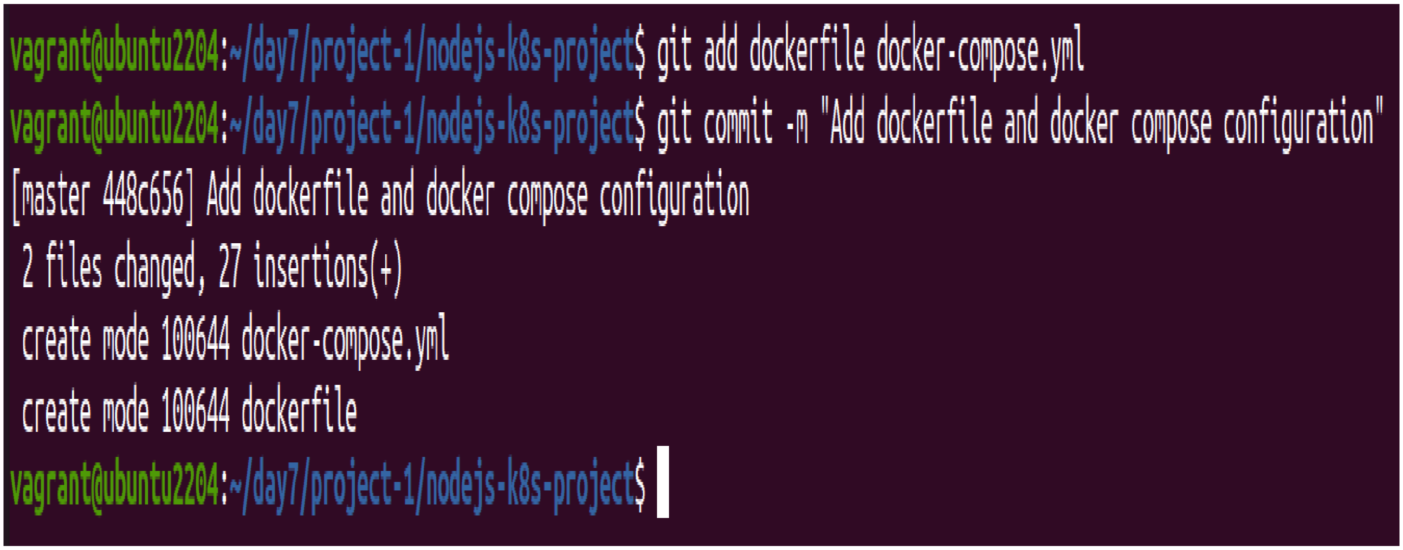
```
version: '3'
services:
 app:
  build: .
  ports:
   - "3000:3000"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ vim docker-compose.yml
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ cat docker-compose.yml
version: '3'
services:
  app:
    build: .
    ports:
      - "3000:3000"

vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

Add and commit changes:

```
git add Dockerfile docker-compose.yml
git commit -m "Add Dockerfile and Docker Compose configuration"
```



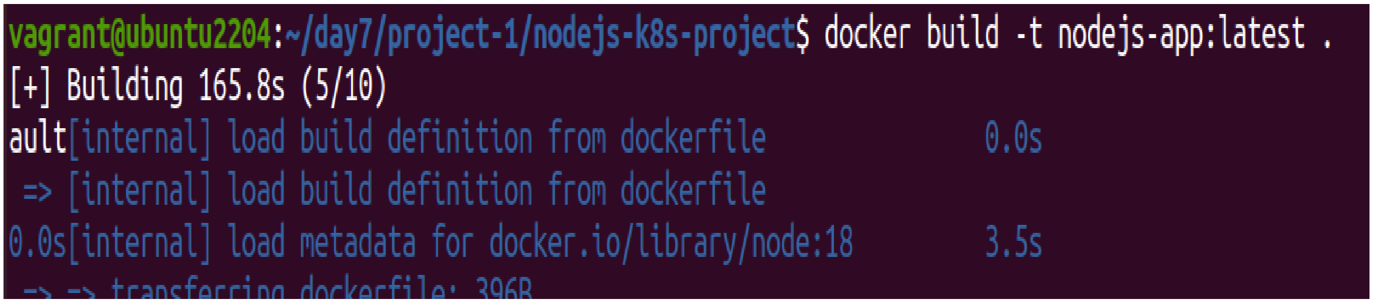
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git add dockerfile docker-compose.yml
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git commit -m "Add dockerfile and docker compose configuration"
[master 448c656] Add dockerfile and docker compose configuration
 2 files changed, 27 insertions(+)
 create mode 100644 docker-compose.yml
 create mode 100644 dockerfile
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

4. Build and Push Docker Image

4.1 Build Docker Image

Build the Docker image:

docker build -t nodejs-app:latest .

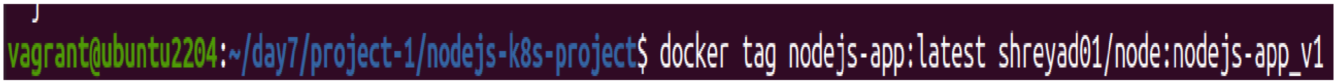


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ docker build -t nodejs-app:latest .
[+] Building 165.8s (5/10)
ault[internal] load build definition from dockerfile              0.0s
 => [internal] load build definition from dockerfile
0.0s[internal] load metadata for docker.io/library/node:18        3.5s
 => => transferring dockerfile: 396B
```

4.2 Push Docker Image to Docker Hub

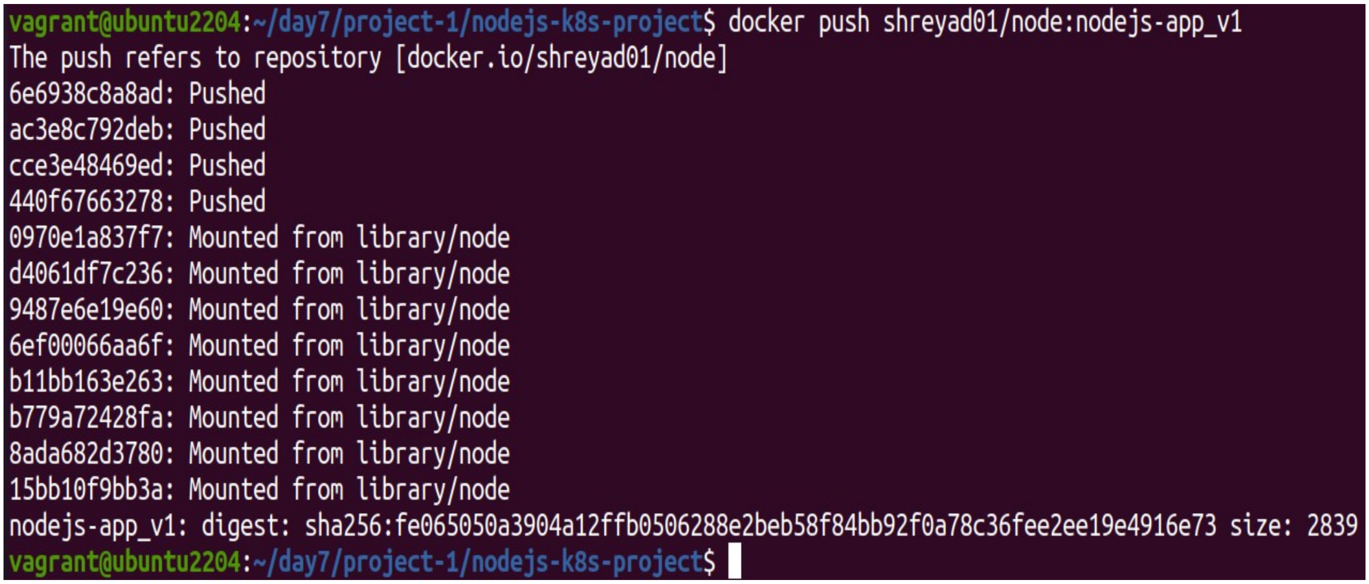
Tag and push the image:

docker tag nodejs-app:latest your-dockerhub-username/nodejs-app:latest



```
 J
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ docker tag nodejs-app:latest shreyad01/node:nodejs-app_v1
```

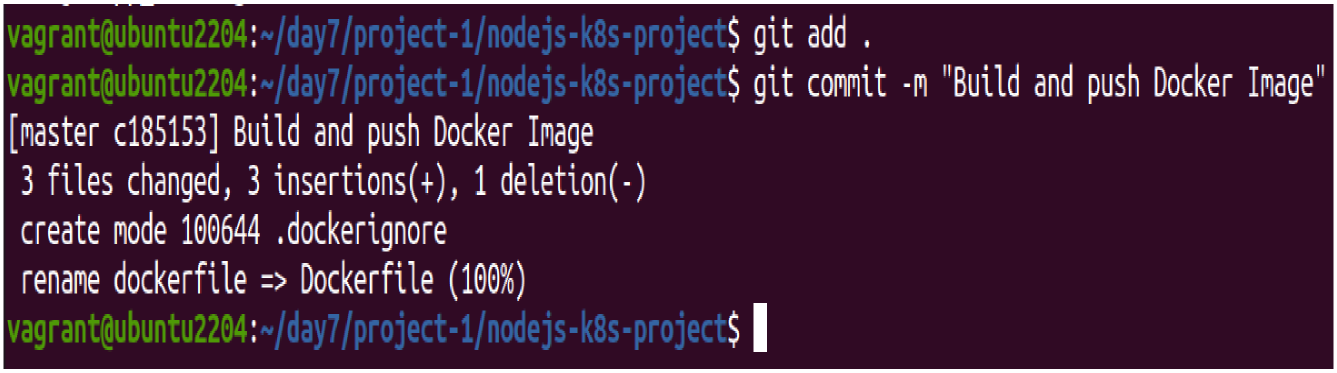
docker push your-dockerhub-username/nodejs-app:latest



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ docker push shreyad01/node:nodejs-app_v1
The push refers to repository [docker.io/shreyad01/node]
6e6938c8a8ad: Pushed
ac3e8c792deb: Pushed
cce3e48469ed: Pushed
440f67663278: Pushed
0970e1a837f7: Mounted from library/node
d4061df7c236: Mounted from library/node
9487e6e19e60: Mounted from library/node
6ef00066aa6f: Mounted from library/node
b11bb163e263: Mounted from library/node
b779a72428fa: Mounted from library/node
8ada682d3780: Mounted from library/node
15bb10f9bb3a: Mounted from library/node
nodejs-app_v1: digest: sha256:fe065050a3904a12ffb0506288e2beb58f84bb92f0a78c36fee2ee19e4916e73 size: 2839
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ █
```

Add and commit changes:

```
git add .
git commit -m "Build and push Docker image"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git add .
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$ git commit -m "Build and push Docker Image"
[master c185153] Build and push Docker Image
 3 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 .dockerignore
 rename dockerfile => Dockerfile (100%)
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project$
```

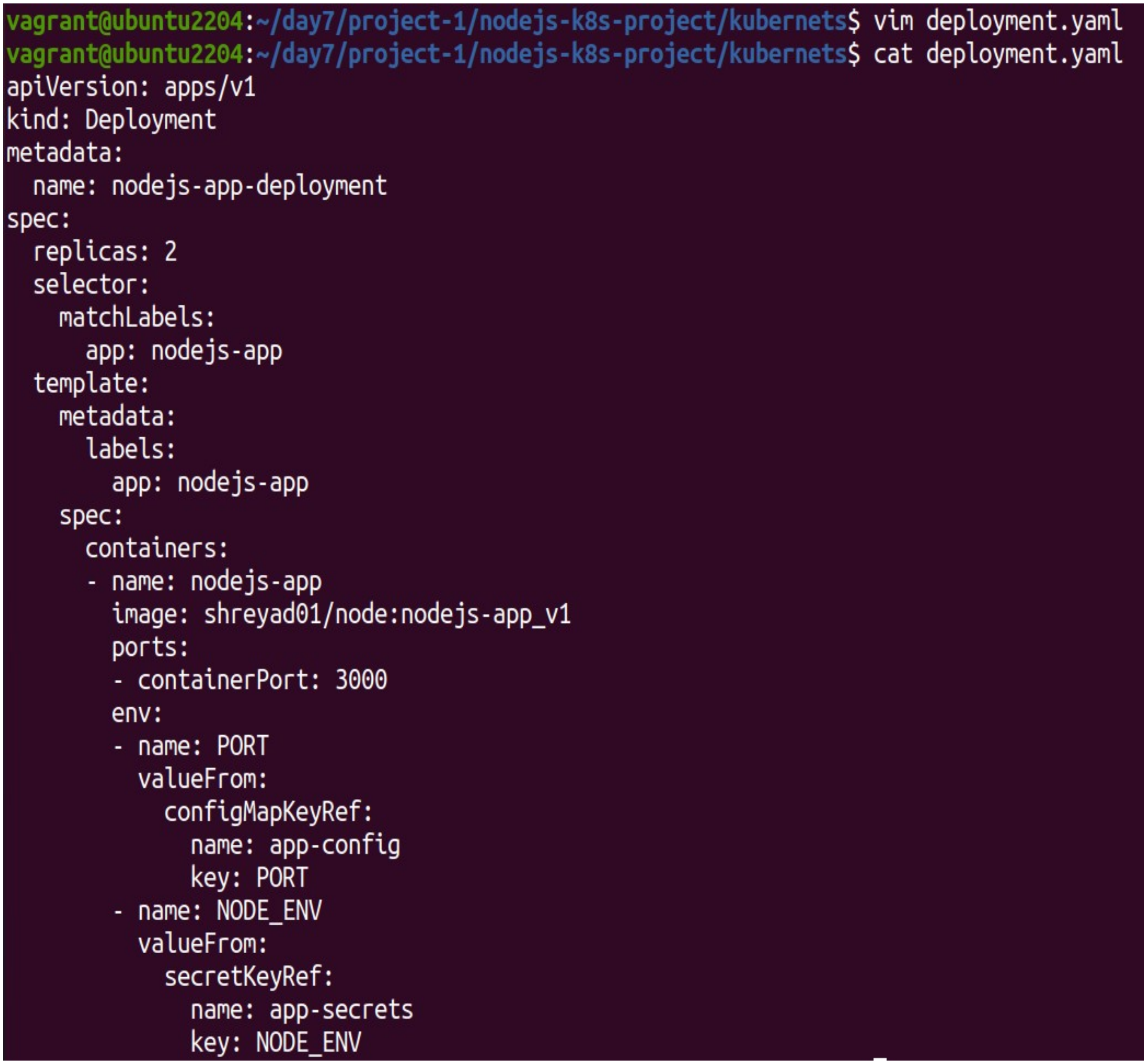
5. Create Kubernetes Configurations

5.1 Create Kubernetes Deployment

Create kubernetes/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:
```

```yaml
- containerPort: 3000
env:
- name: PORT
  valueFrom:
    configMapKeyRef:
      name: app-config
      key: PORT
- name: NODE_ENV
  valueFrom:
    secretKeyRef:
      name: app-secrets
      key: NODE_ENV
```

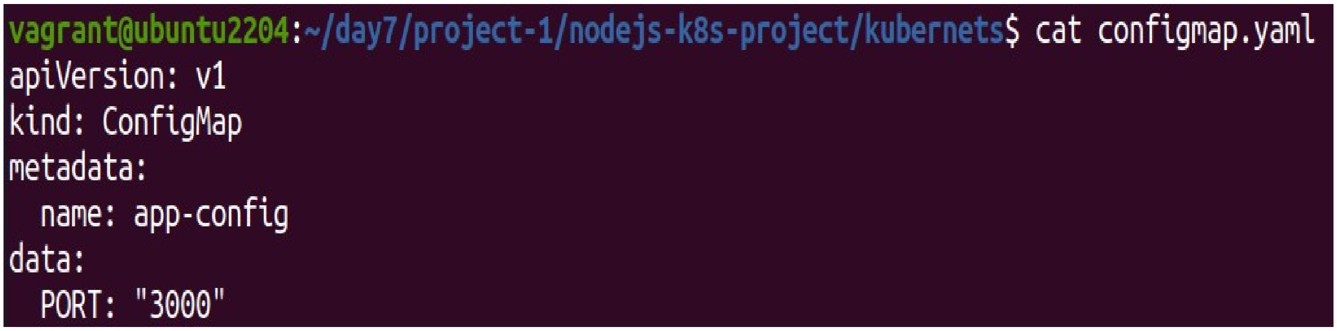


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ vim deployment.yaml
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - name: nodejs-app
        image: shreyad01/node:nodejs-app_v1
        ports:
        - containerPort: 3000
        env:
        - name: PORT
          valueFrom:
            configMapKeyRef:
              name: app-config
              key: PORT
        - name: NODE_ENV
          valueFrom:
            secretKeyRef:
              name: app-secrets
              key: NODE_ENV
```

5.2 Create ConfigMap and Secret

Create kubernetes/configmap.yaml:

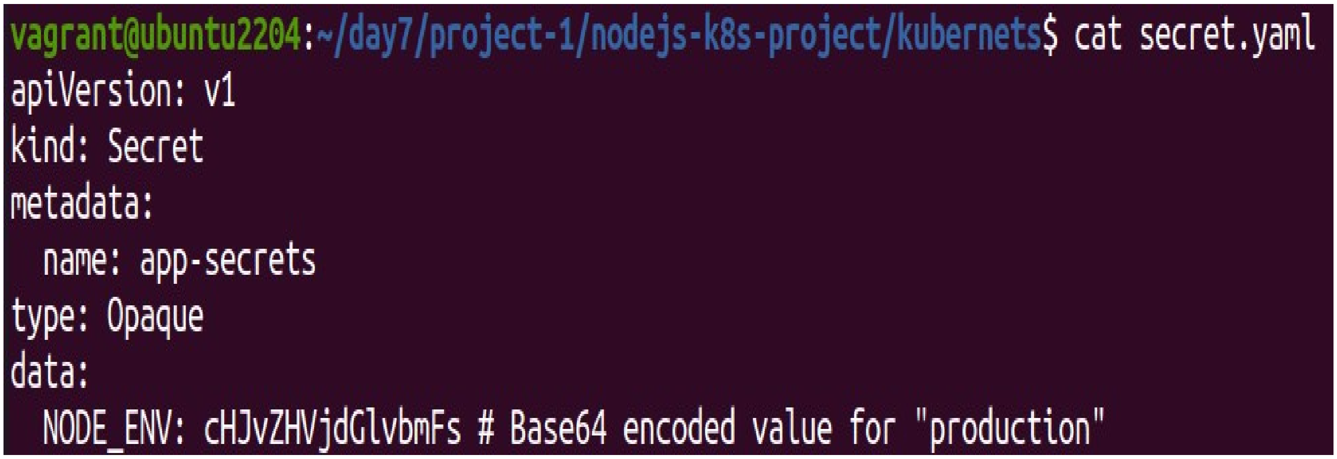
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
 data:
  PORT: "3000"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"
```

Create kubernetes/secret.yaml:

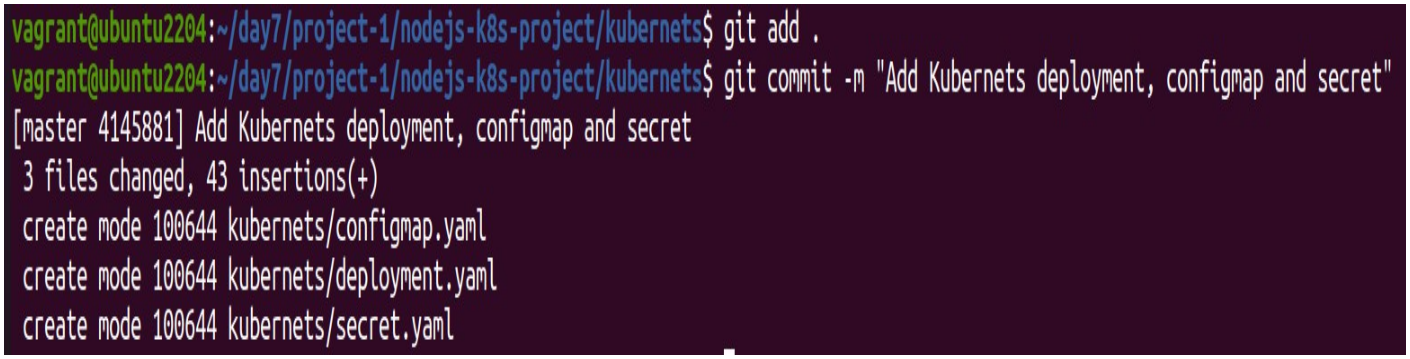
```
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbmFs # Base64 encoded value for "production"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
  NODE_ENV: cHJvZHVjdGlvbmFs # Base64 encoded value for "production"
```

Add and commit Kubernetes configurations:

```
git add kubernetes/
git commit -m "Add Kubernetes deployment, configmap, and secret"
```

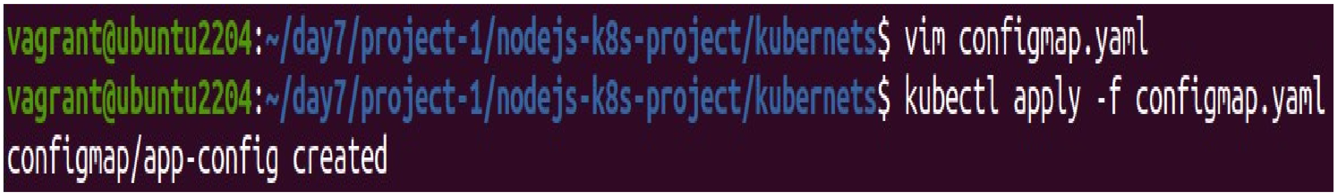


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git add .
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git commit -m "Add Kubernets deployment, configmap and secret"
[master 4145881] Add Kubernets deployment, configmap and secret
 3 files changed, 43 insertions(+)
 create mode 100644 kubernets/configmap.yaml
 create mode 100644 kubernets/deployment.yaml
 create mode 100644 kubernets/secret.yaml
```

5.3 Apply Kubernetes Configurations

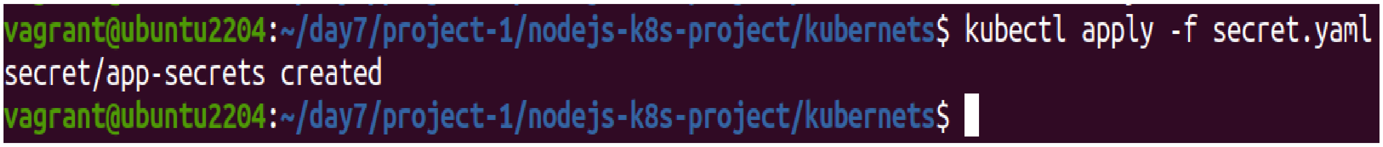
Apply the ConfigMap and Secret:

kubectl apply -f kubernetes/configmap.yaml



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ vim configmap.yaml
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl apply -f configmap.yaml
configmap/app-config created
```

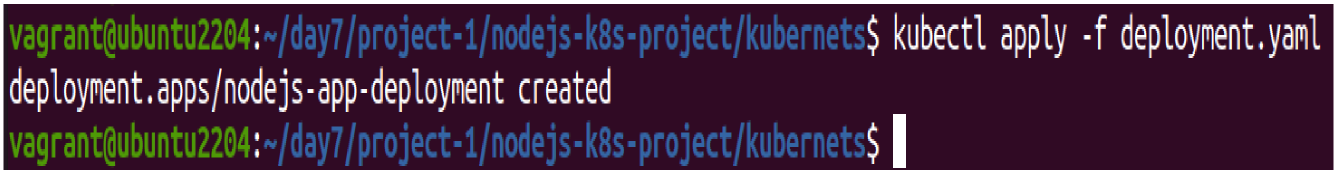
kubectl apply -f kubernetes/secret.yaml



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl apply -f secret.yaml
secret/app-secrets created
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$
```

Apply the Deployment:

kubectl apply -f kubernetes/deployment.yaml



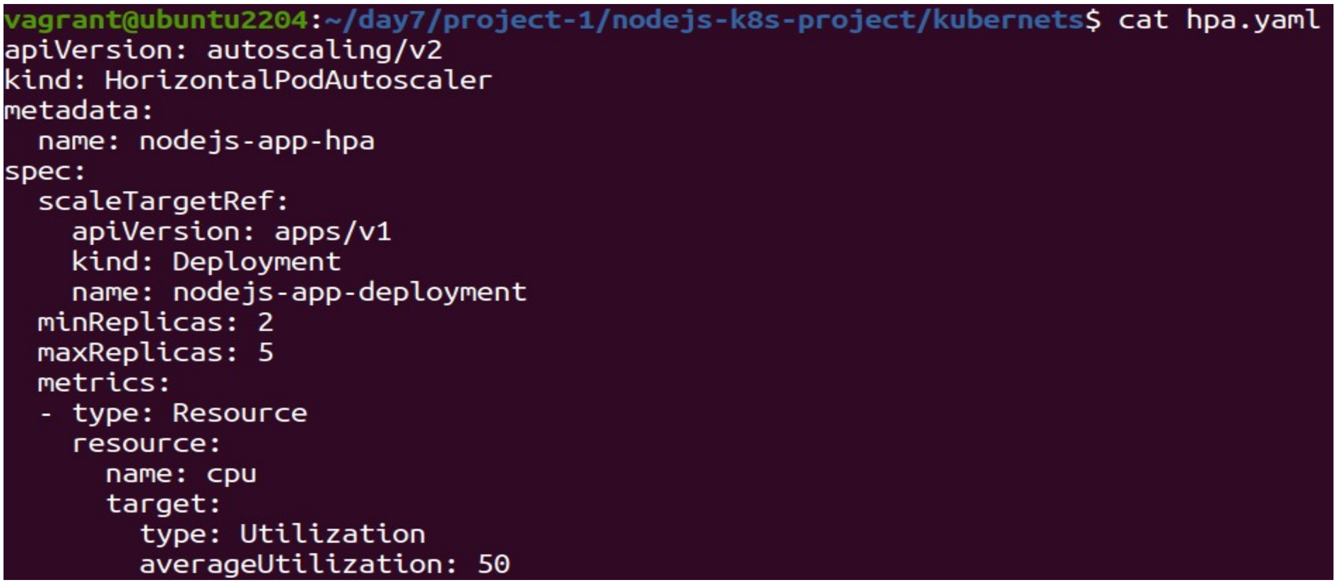
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app-deployment created
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$
```

6. Implement Autoscaling

6.1 Create Horizontal Pod Autoscaler

Create kubernetes/hpa.yaml:

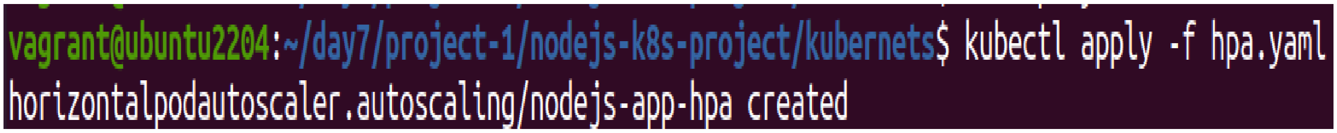
```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-app-deployment
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target: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



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ cat hpa.yaml
apiVersion: autoscaling/v2
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
```

Apply the HPA:

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



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-app-hpa created
```

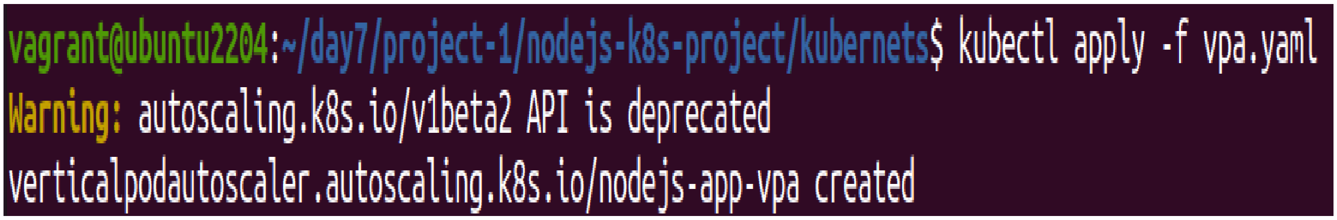
6.2 Create Vertical Pod Autoscaler

Create kubernetes/vpa.yaml:

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

Apply the VPA:

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



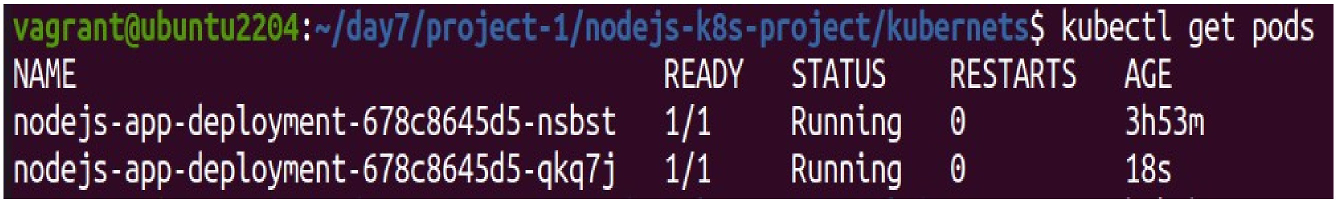
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl apply -f vpa.yaml
Warning: autoscaling.k8s.io/v1beta2 API is deprecated
verticalpodautoscaler.autoscaling.k8s.io/nodejs-app-vpa created
```

7. Test the Deployment

7.1 Check the Status of Pods, Services, and HPA

Verify the Pods:

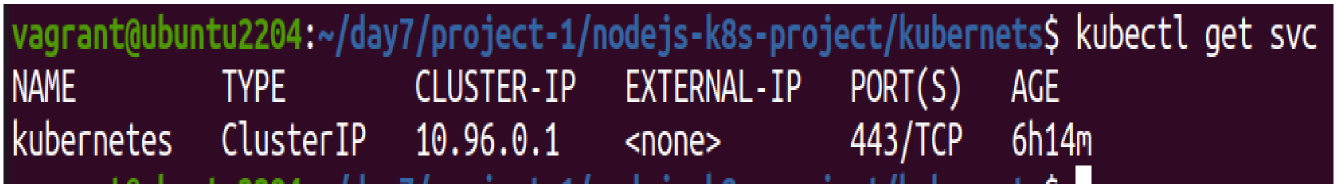
kubectl get pods



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
nodejs-app-deployment-678c8645d5-nsbst  1/1     Running   0          3h53m
nodejs-app-deployment-678c8645d5-qkq7j  1/1     Running   0          18s
```

Verify the Services:

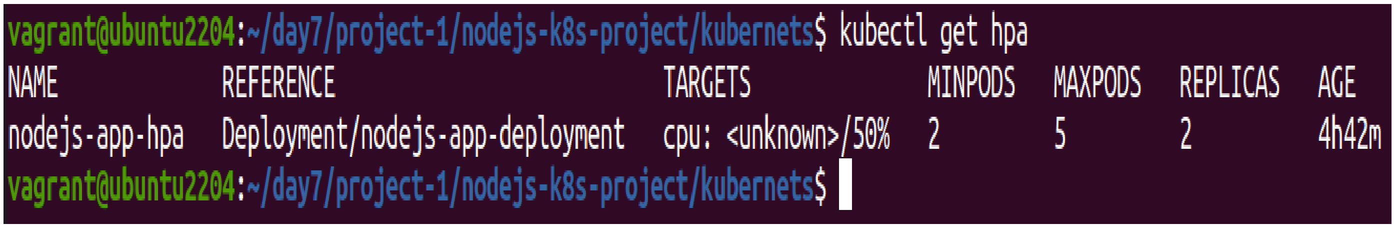
kubectl get svc



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   6h14m
```

Verify the HPA:

kubectl get hpa

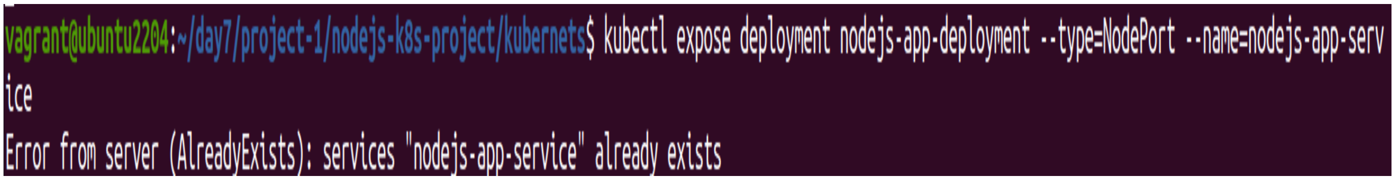


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl get hpa
NAME             REFERENCE                          TARGETS             MINPODS   MAXPODS   REPLICAS   AGE
nodejs-app-hpa   Deployment/nodejs-app-deployment   cpu: <unknown>/50%  2         5         2          4h42m
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$
```

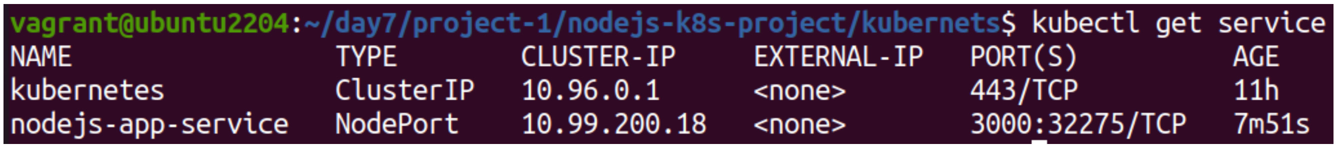
7.2 Access the Application

Expose the Service:

kubectl expose deployment nodejs-app-deployment --type=NodePort –name=nodejs-app-service



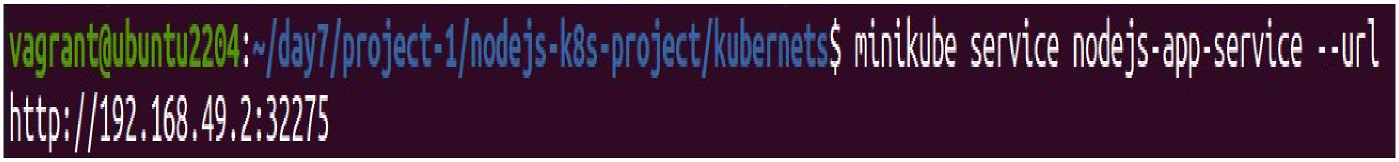
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl expose deployment nodejs-app-deployment --type=NodePort --name=nodejs-app-serv
ice
Error from server (AlreadyExists): services "nodejs-app-service" already exists
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ kubectl get service
NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP          11h
nodejs-app-service   NodePort    10.99.200.18   <none>        3000:32275/TCP   7m51s
```

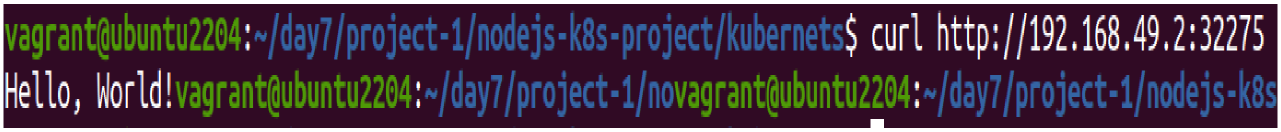
Get the Minikube IP and Service Port:

minikube service nodejs-app-service –url



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ minikube service nodejs-app-service --url
http://192.168.49.2:32275
```

- **Access the Application** in your browser using the URL obtained from the previous command.



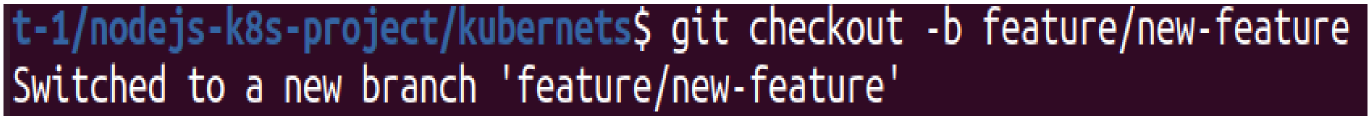
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ curl http://192.168.49.2:32275
Hello, World!vagrant@ubuntu2204:~/day7/project-1/novagrant@ubuntu2204:~/day7/project-1/nodejs-k8s
```

8. Git Version Control

8.1 Create a New Branch for New Features

Create and switch to a new branch:

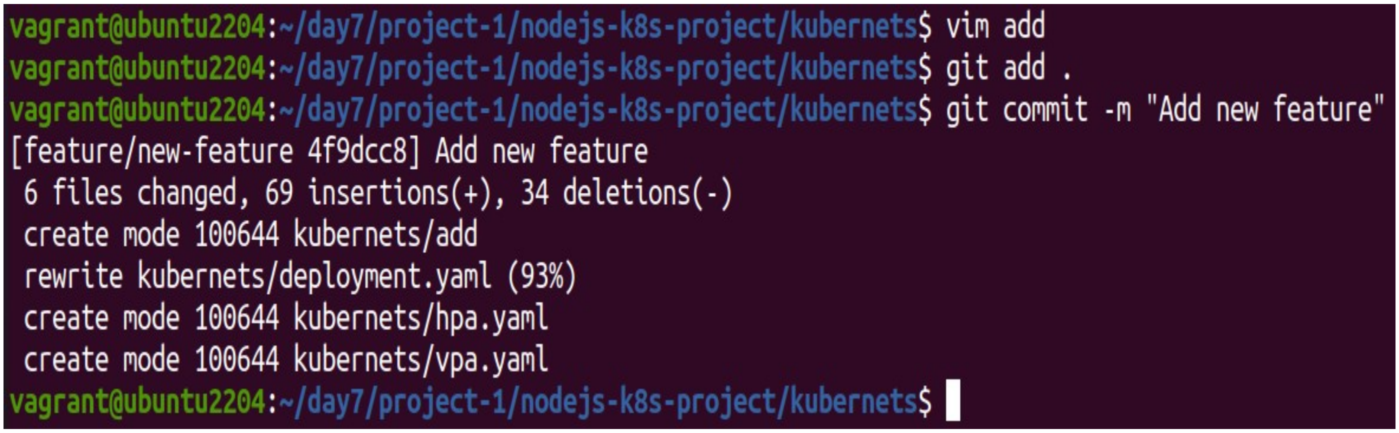
git checkout -b feature/new-feature



```
t-1/nodejs-k8s-project/kubernets$ git checkout -b feature/new-feature
Switched to a new branch 'feature/new-feature'
```

Make changes and commit:

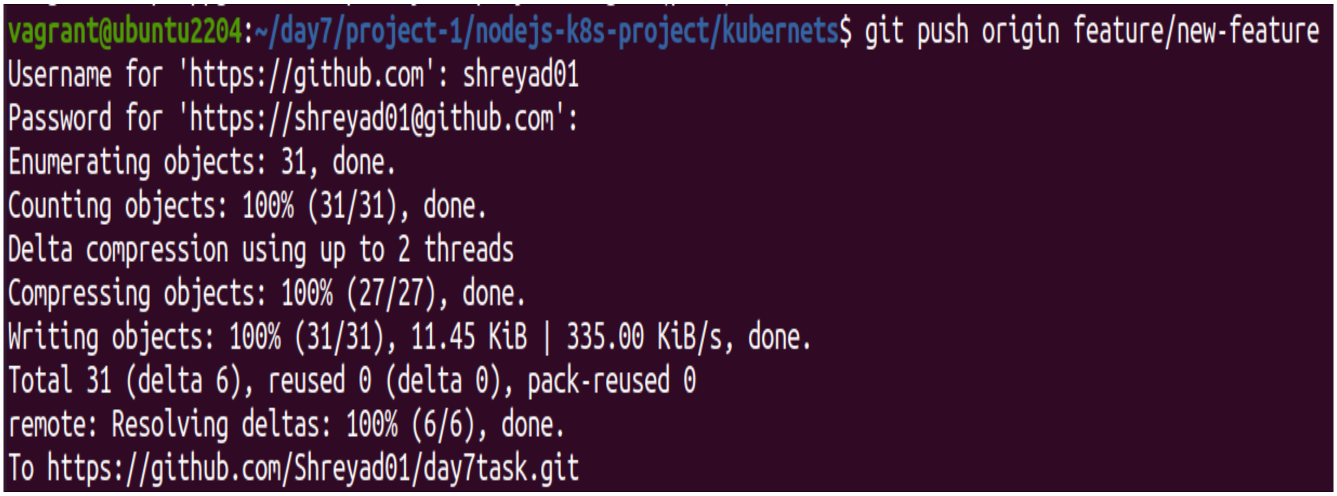
```
# Make some changes
git add .
git commit -m "Add new feature"
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ vim add
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git add .
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git commit -m "Add new feature"
[feature/new-feature 4f9dcc8] Add new feature
 6 files changed, 69 insertions(+), 34 deletions(-)
 create mode 100644 kubernets/add
 rewrite kubernets/deployment.yaml (93%)
 create mode 100644 kubernets/hpa.yaml
 create mode 100644 kubernets/vpa.yaml
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ 
```

Push the branch to the remote repository:

```
git push origin feature/new-feature
```

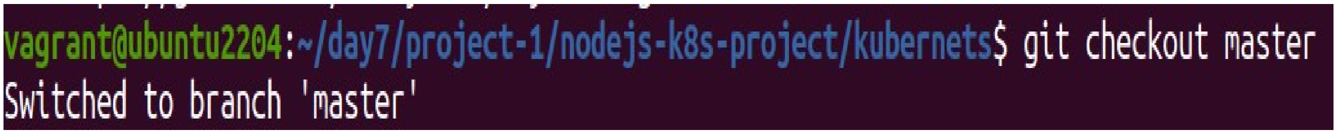


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git push origin feature/new-feature
Username for 'https://github.com': shreyad01
Password for 'https://shreyad01@github.com':
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 2 threads
Compressing objects: 100% (27/27), done.
Writing objects: 100% (31/31), 11.45 KiB | 335.00 KiB/s, done.
Total 31 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/Shreyad01/day7task.git
```

8.2 Rebase Feature Branch on Main Branch

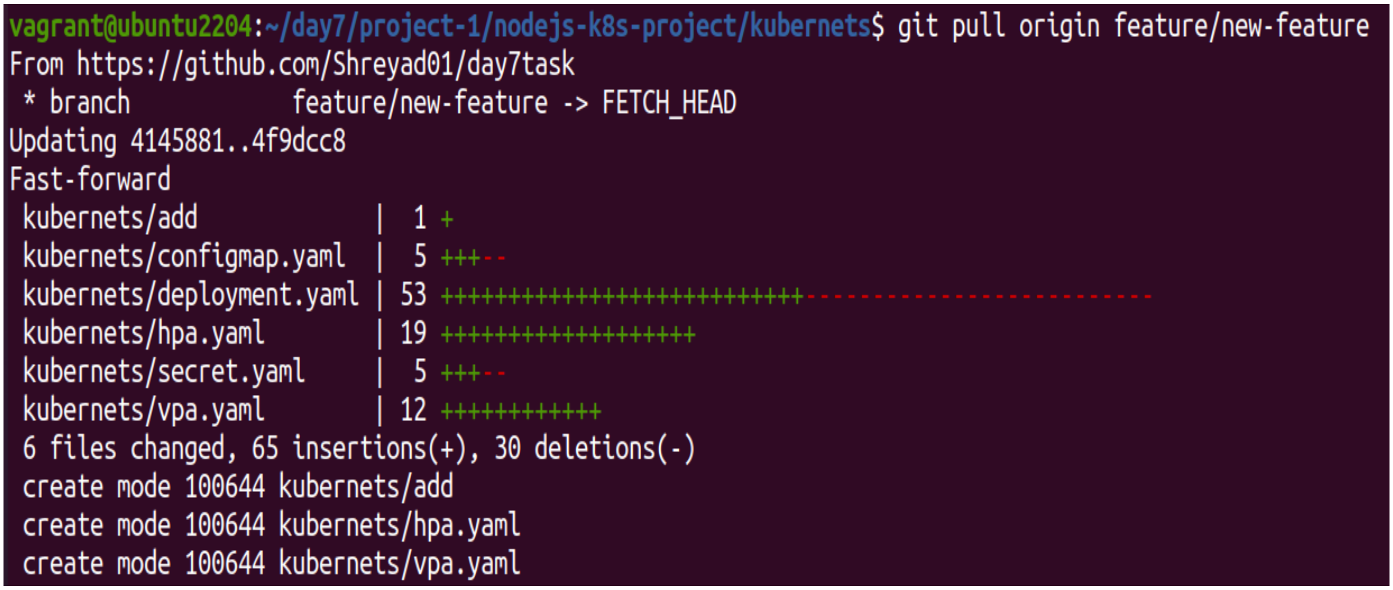
Switch to the main branch and pull the latest changes:

```
git checkout main
```



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git checkout master
Switched to branch 'master'
```

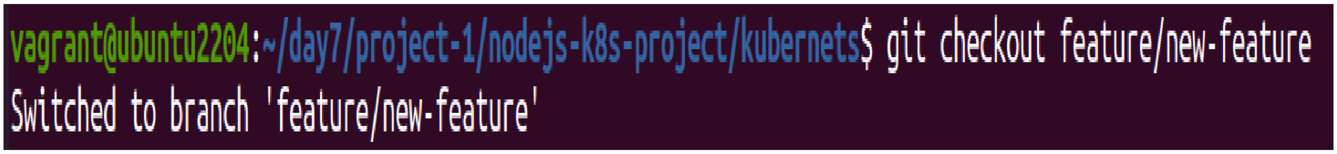
git pull origin main



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git pull origin feature/new-feature
From https://github.com/Shreyad01/day7task
 * branch            feature/new-feature -> FETCH_HEAD
Updating 4145881..4f9dcc8
Fast-forward
 kubernets/add             |  1 +
 kubernets/configmap.yaml  |  5 +++--
 kubernets/deployment.yaml | 53 +++++++++++++++++++++++++++--------------------------
 kubernets/hpa.yaml        | 19 +++++++++++++++++++
 kubernets/secret.yaml     |  5 +++--
 kubernets/vpa.yaml        | 12 ++++++++++++
 6 files changed, 65 insertions(+), 30 deletions(-)
 create mode 100644 kubernets/add
 create mode 100644 kubernets/hpa.yaml
 create mode 100644 kubernets/vpa.yaml
```

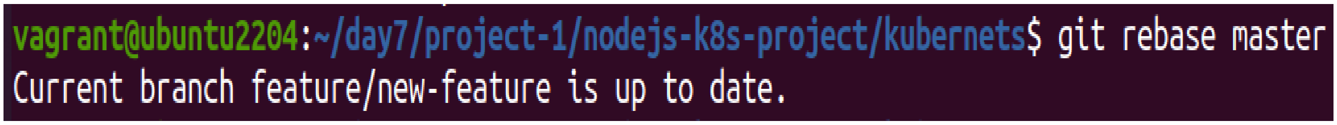
Rebase the feature branch:

git checkout feature/new-feature



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git checkout feature/new-feature
Switched to branch 'feature/new-feature'
```

git rebase main



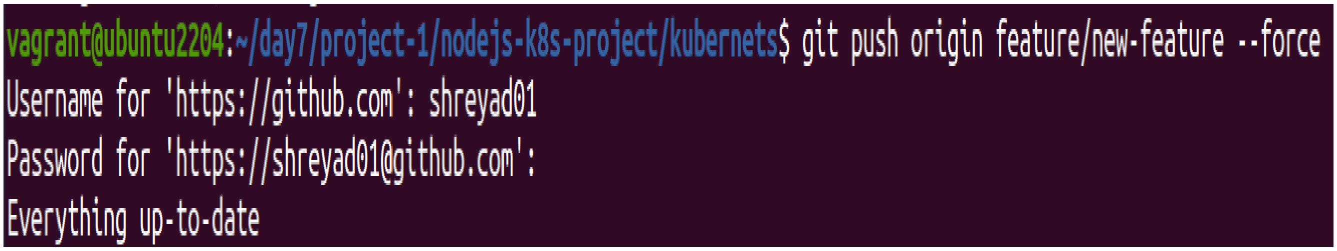
```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git rebase master
Current branch feature/new-feature is up to date.
```

Resolve conflicts if any, and continue the rebase:

git add .
git rebase --continue

Push the rebased feature branch:

git push origin feature/new-feature –force

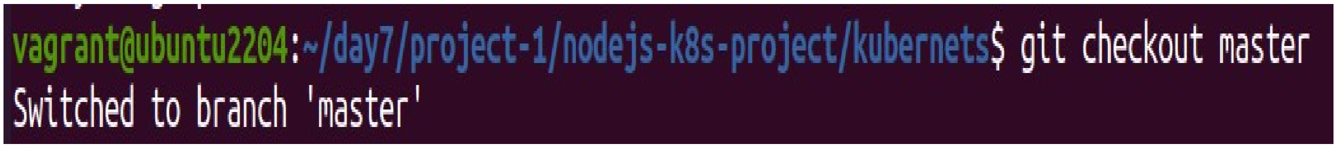


```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git push origin feature/new-feature --force
Username for 'https://github.com': shreyad01
Password for 'https://shreyad01@github.com':
Everything up-to-date
```

9. Final Commit and Cleanup

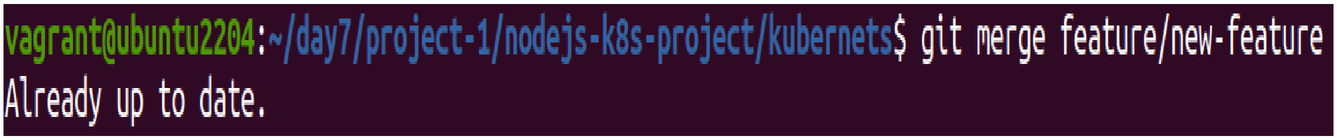
Merge feature branch to main:

git checkout main



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git checkout master
Switched to branch 'master'
```

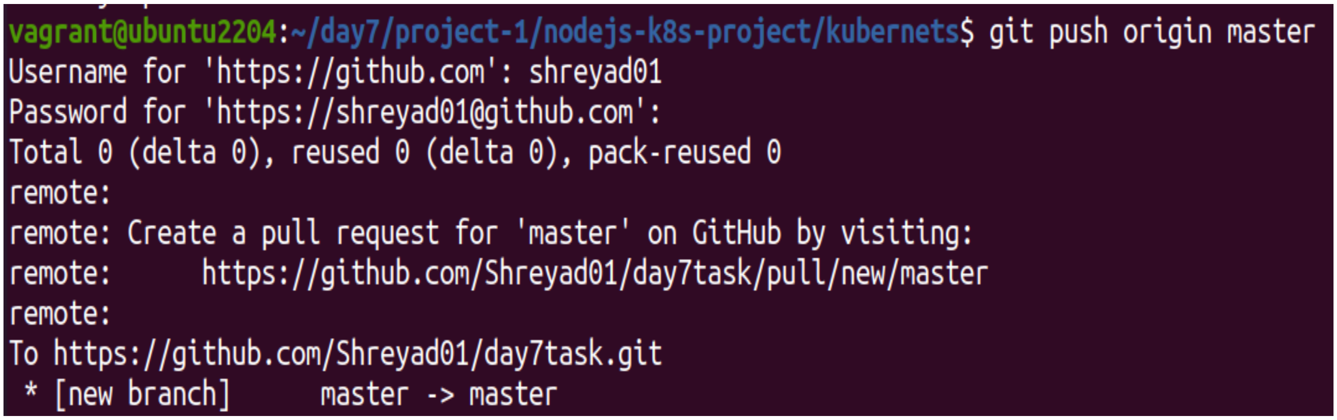
git merge feature/new-feature



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git merge feature/new-feature
Already up to date.
```

Push the changes to the main branch:

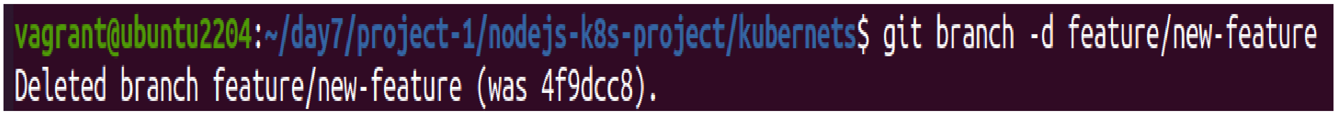
git push origin main



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git push origin master
Username for 'https://github.com': shreyad01
Password for 'https://shreyad01@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Shreyad01/day7task/pull/new/master
remote:
To https://github.com/Shreyad01/day7task.git
 * [new branch]      master -> master
```

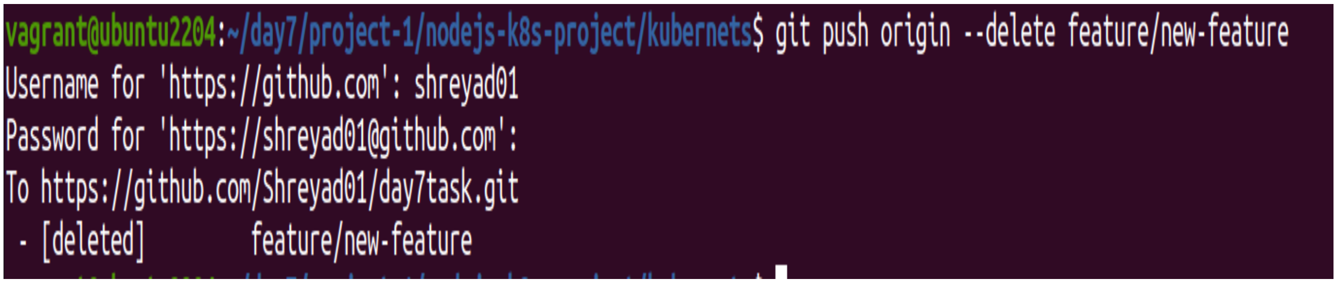
Clean up:

git branch -d feature/new-feature



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git branch -d feature/new-feature
Deleted branch feature/new-feature (was 4f9dcc8).
```

git push origin --delete feature/new-feature



```
vagrant@ubuntu2204:~/day7/project-1/nodejs-k8s-project/kubernets$ git push origin --delete feature/new-feature
Username for 'https://github.com': shreyad01
Password for 'https://shreyad01@github.com':
To https://github.com/Shreyad01/day7task.git
 - [deleted]         feature/new-feature
```

Project 02

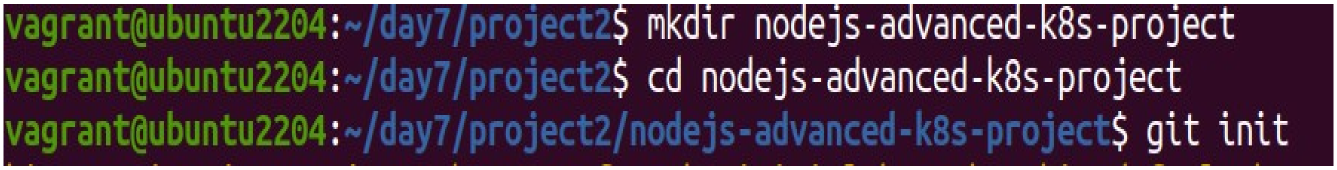
Deploy a Node.js application to Kubernetes with advanced usage of ConfigMaps and Secrets. Implement Horizontal Pod Autoscaler (HPA) with both scale-up and scale-down policies. The project will include a multi-environment configuration strategy, integrating a Redis cache, and monitoring application metrics.

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
```

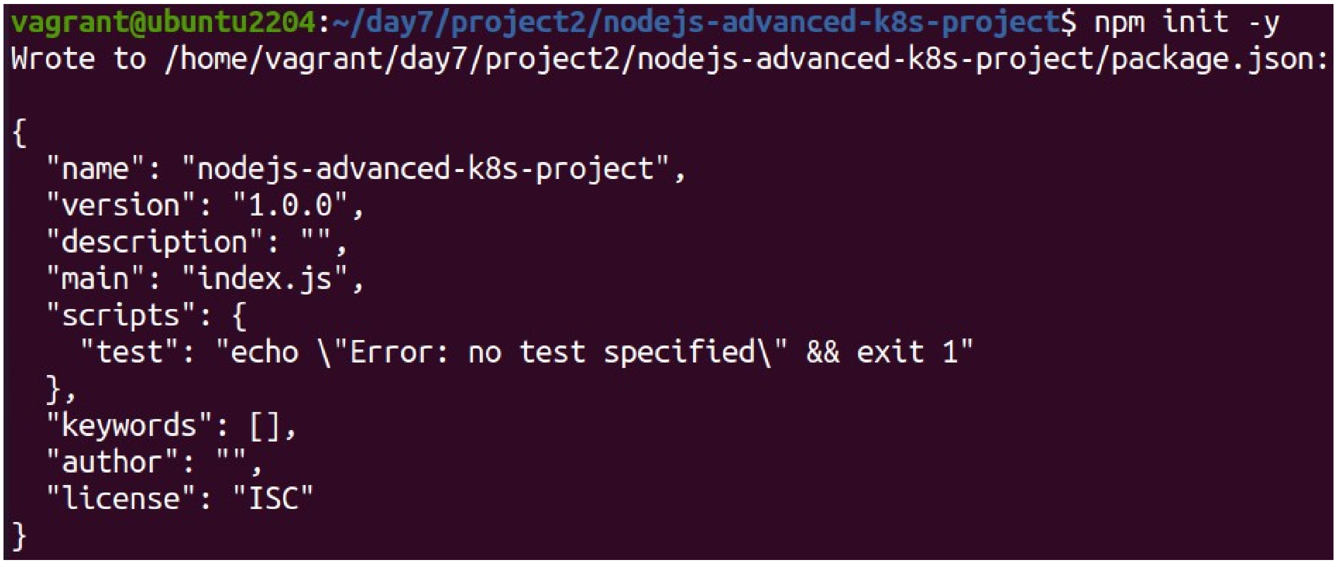


```
vagrant@ubuntu2204:~/day7/project2$ mkdir nodejs-advanced-k8s-project
vagrant@ubuntu2204:~/day7/project2$ cd nodejs-advanced-k8s-project
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ git init
```

1.2 Create Initial Files

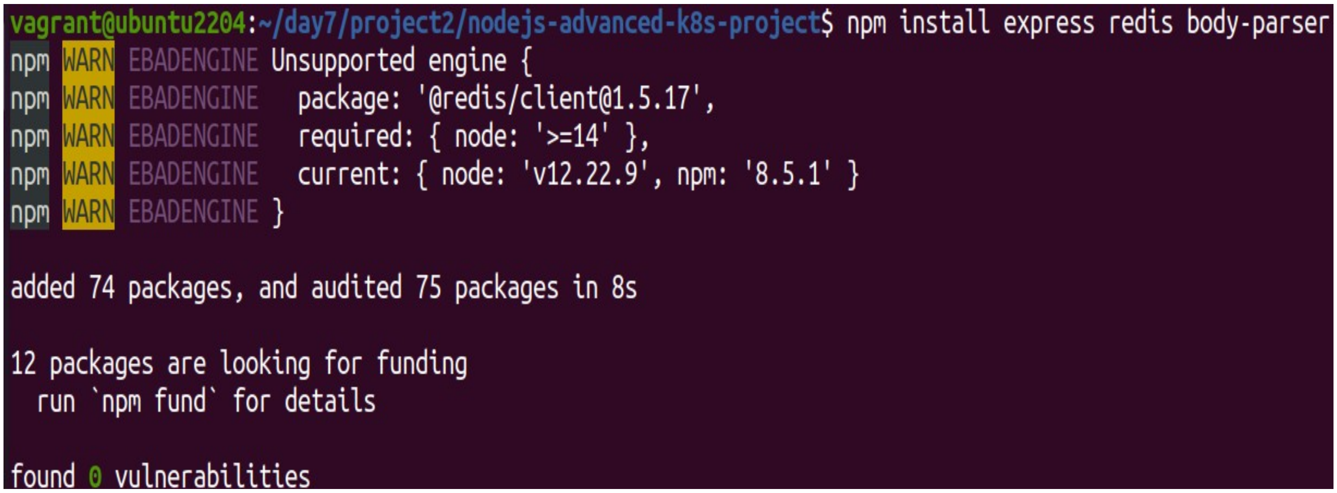
Create the initial Node.js application and Docker-related files:

npm init -y



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ npm init -y
Wrote to /home/vagrant/day7/project2/nodejs-advanced-k8s-project/package.json:

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

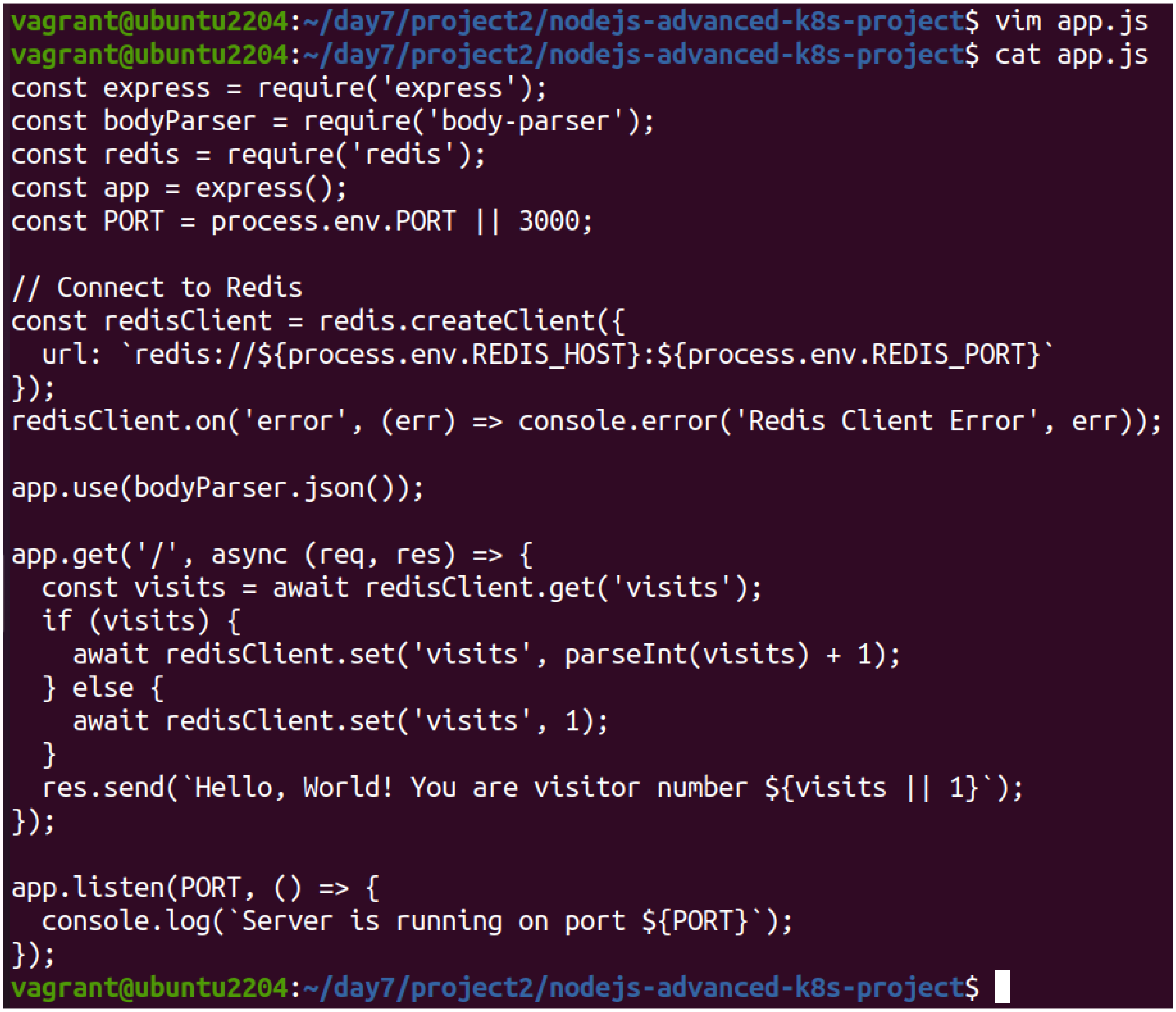
npm install express redis body-parser



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ npm install express redis body-parser
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@redis/client@1.5.17',
npm WARN EBADENGINE   required: { node: '>=14' },
npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
npm WARN EBADENGINE }

added 74 packages, and audited 75 packages in 8s

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

found 0 vulnerabilities
```

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}`);
});
```

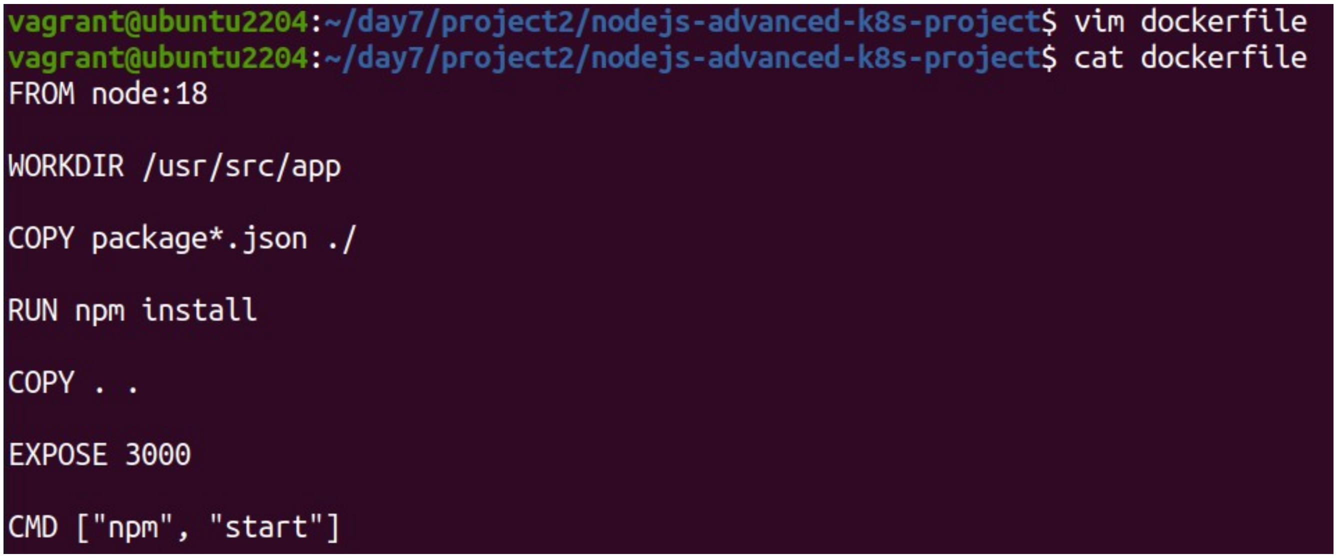


```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ vim app.js
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ cat app.js
const express = require('express');
const bodyParser = require('body-parser');
const redis = require('redis');
const app = express();
const PORT = 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 ${PORT}`);
});
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$
```

Dockerfile

```
FROM node:18

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install
```

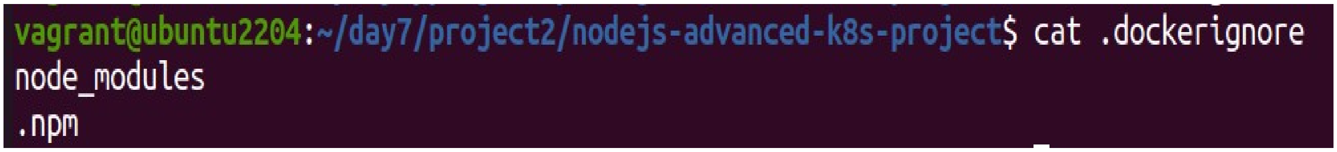
```
COPY . .

EXPOSE 3000

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



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ vim dockerfile
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ cat dockerfile
FROM node:18

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

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

.dockerignore

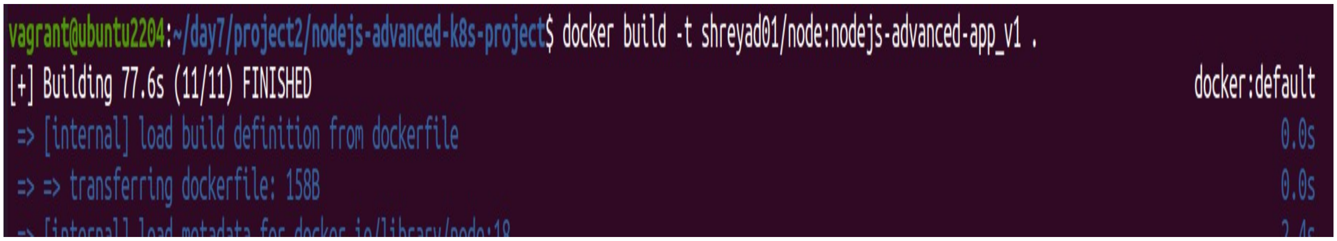
```
node_modules
.npm
```



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ cat .dockerignore
node_modules
.npm
```

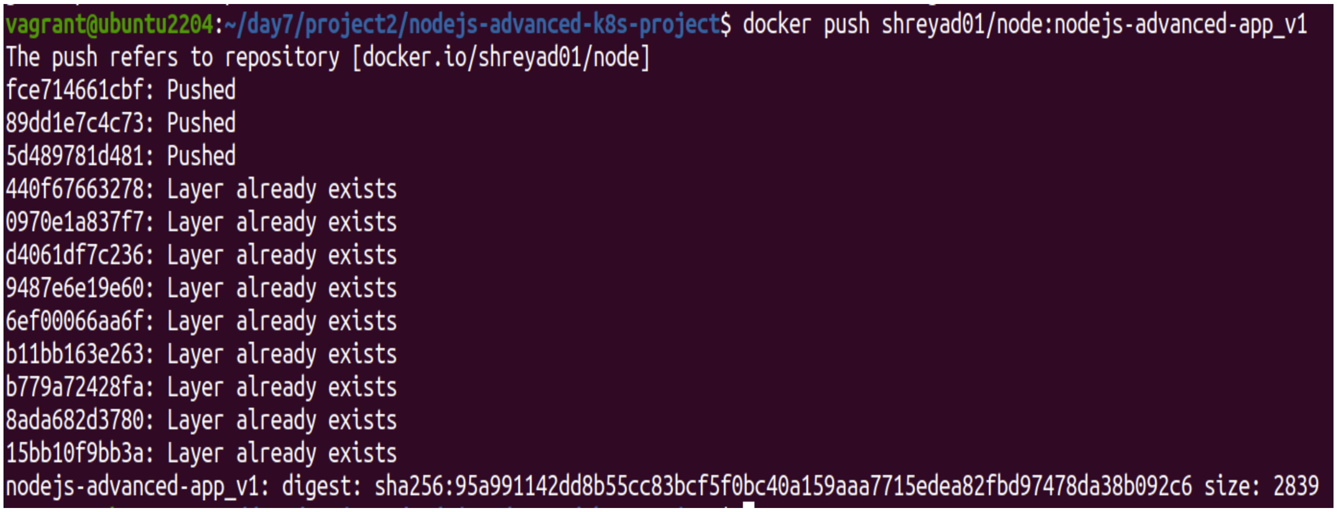
1. Build and push Docker image:

```
docker build -t your-dockerhub-username/nodejs-advanced-app:latest .
```



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ docker build -t shreyad01/node:nodejs-advanced-app_v1 .
[+] Building 77.6s (11/11) FINISHED                                              docker:default
 => [internal] load build definition from dockerfile                                       0.0s
 => => transferring dockerfile: 158B                                                       0.0s
 => [internal] load metadata for docker.io/library/node:18                                 2.4s
```

```
  docker push your-dockerhub-username/nodejs-advanced-app:latest
```



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project$ docker push shreyad01/node:nodejs-advanced-app_v1
The push refers to repository [docker.io/shreyad01/node]
fce714661cbf: Pushed
89dd1e7c4c73: Pushed
5d489781d481: Pushed
440f67663278: Layer already exists
0970e1a837f7: Layer already exists
d4061df7c236: Layer already exists
9487e6e19e60: Layer already exists
6ef00066aa6f: Layer already exists
b11bb163e263: Layer already exists
b779a72428fa: Layer already exists
8ada682d3780: Layer already exists
15bb10f9bb3a: Layer already exists
nodejs-advanced-app_v1: digest: sha256:95a991142dd8b55cc83bcf5f0bc40a159aaa7715edea82fbd97478da38b092c6 size: 2839
```

2. Advanced Kubernetes Configuration

2.1 Deployment Configuration

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

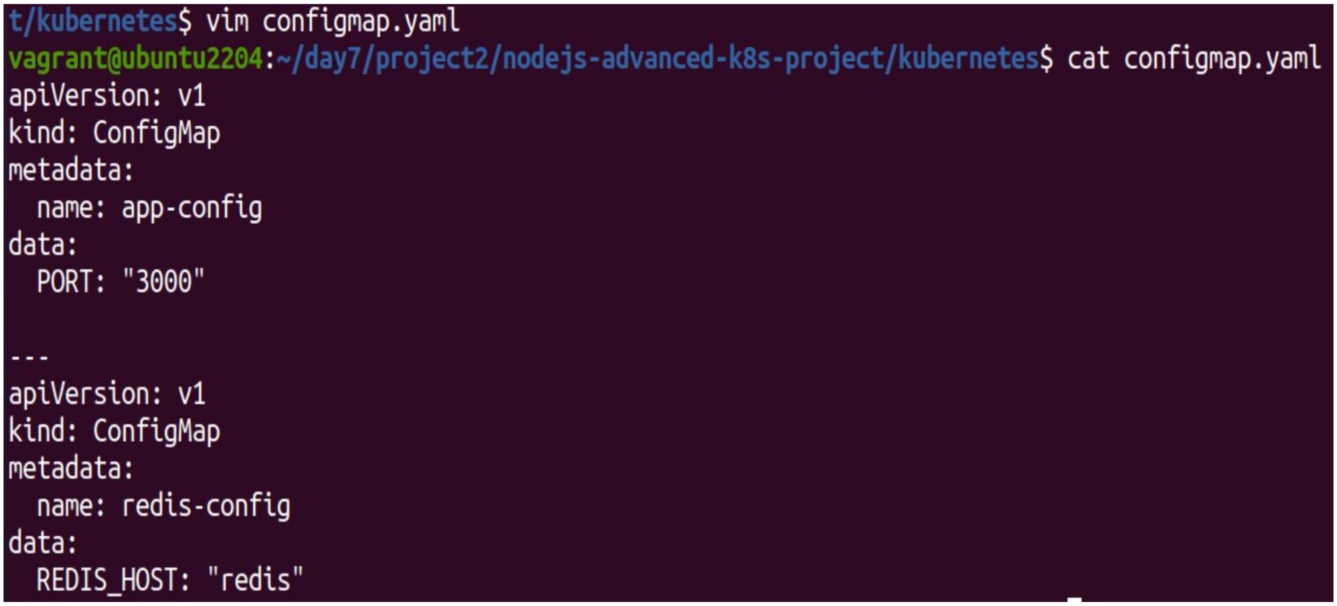
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-advanced-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-advanced-app
  template:
    metadata:
      labels:
        app: nodejs-advanced-app
    spec:
      containers:
```

```
    - 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"
```

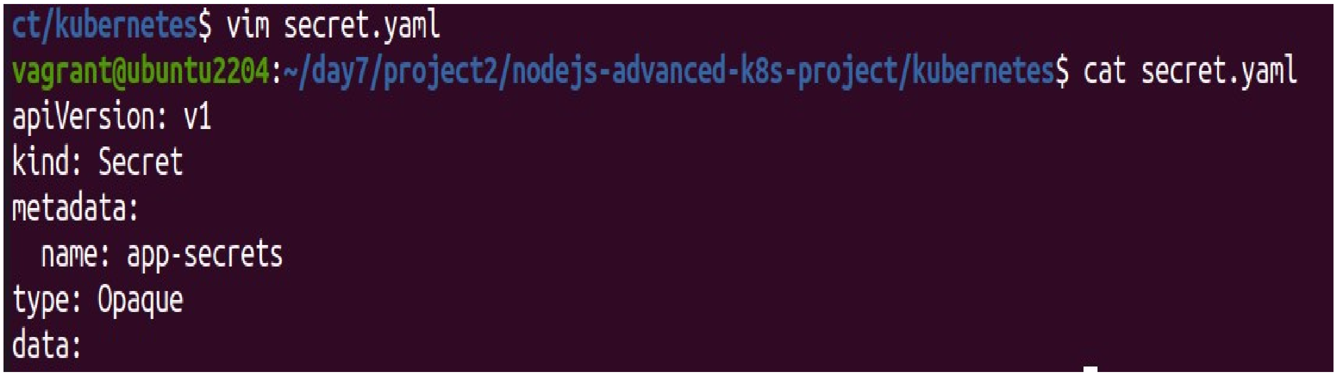


```
t/kubernetes$ vim configmap.yaml
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  PORT: "3000"

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

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"
```

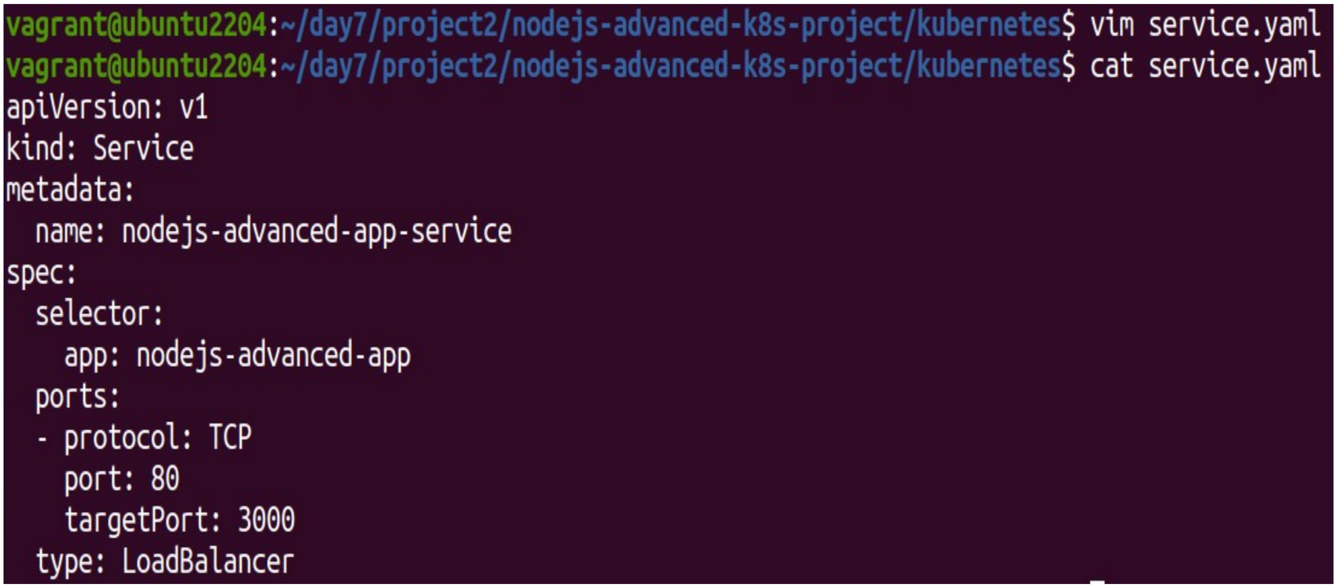


```
ct/kubernetes$ vim secret.yaml
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secrets
type: Opaque
data:
```

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
```

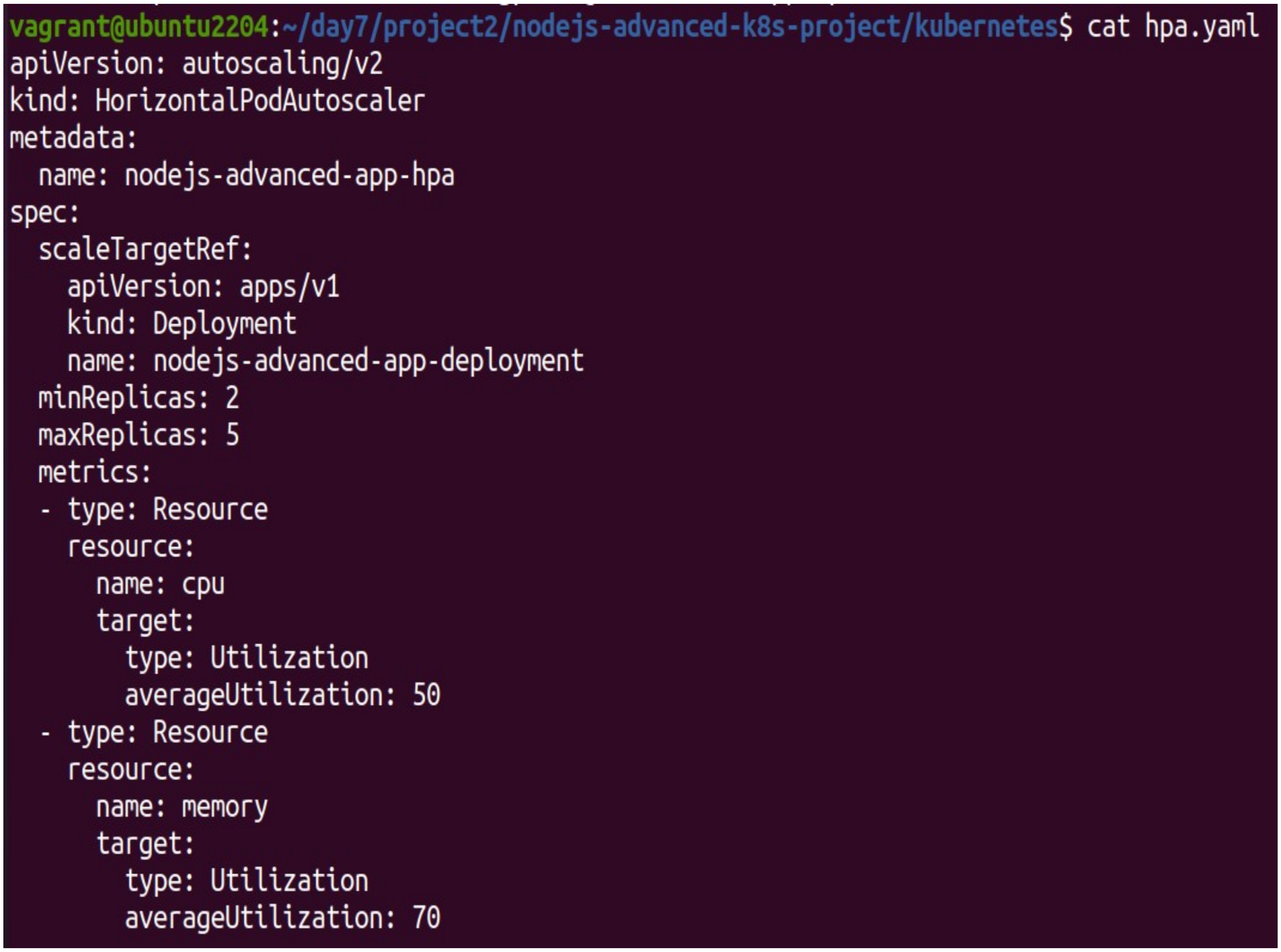


```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ vim service.yaml
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nodejs-advanced-app-service
spec:
  selector:
    app: nodejs-advanced-app
  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/v2beta2
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
```

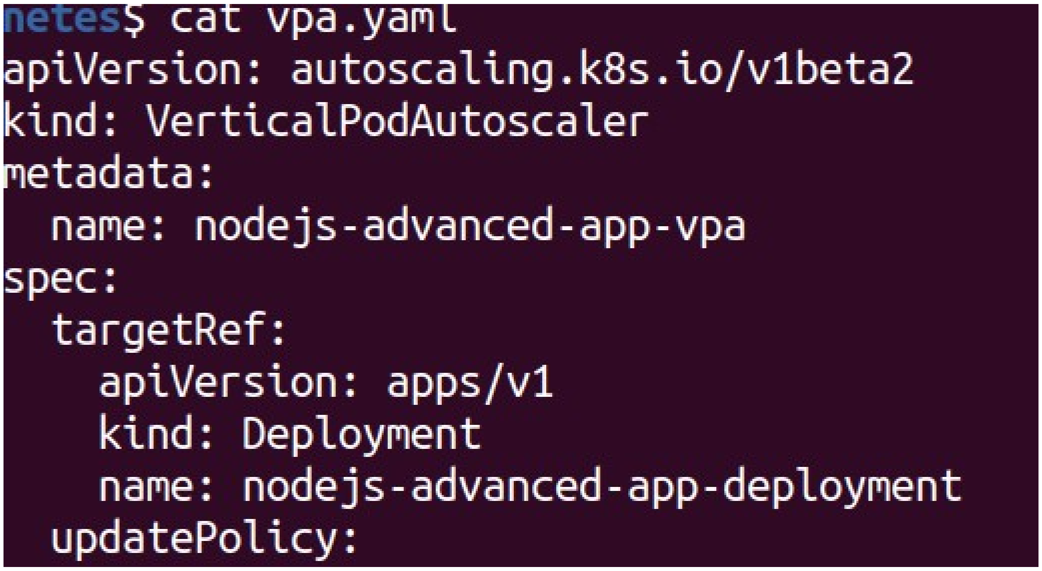


```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat hpa.yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-advanced-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-advanced-app-deployment
  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
```

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"
```

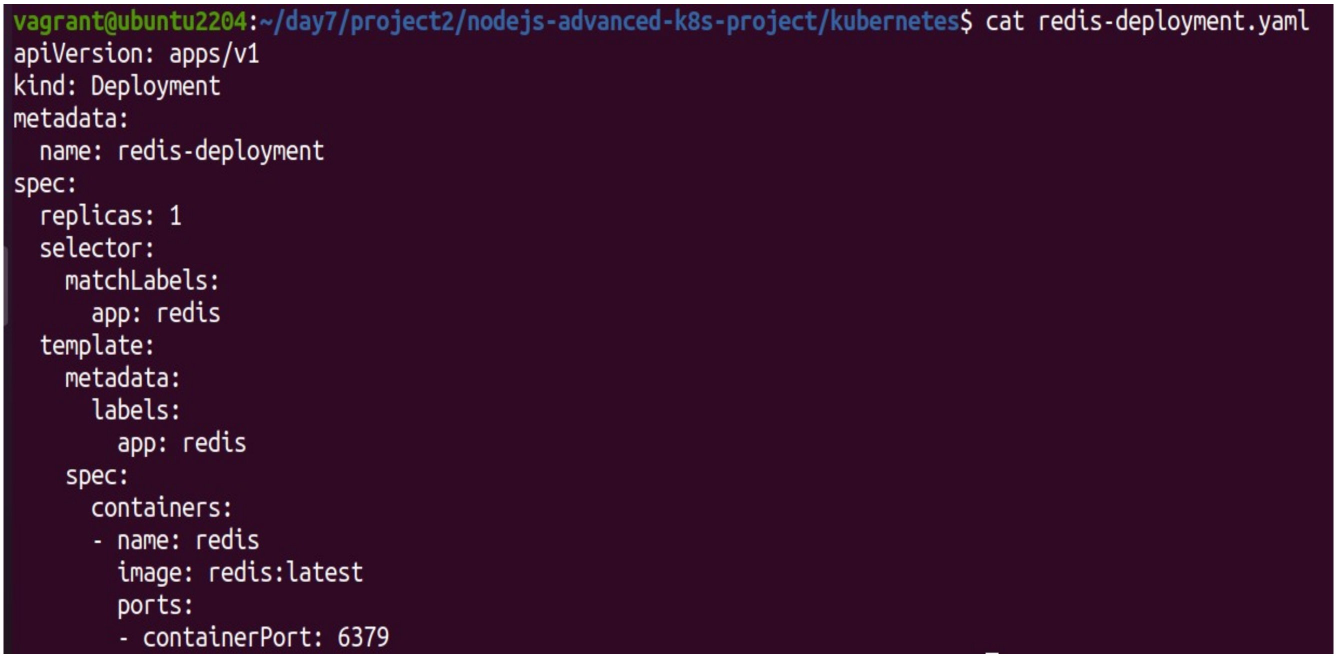


```
netes$ cat vpa.yaml
apiVersion: autoscaling.k8s.io/v1beta2
kind: VerticalPodAutoscaler
metadata:
  name: nodejs-advanced-app-vpa
spec:
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-advanced-app-deployment
  updatePolicy:
```

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
```

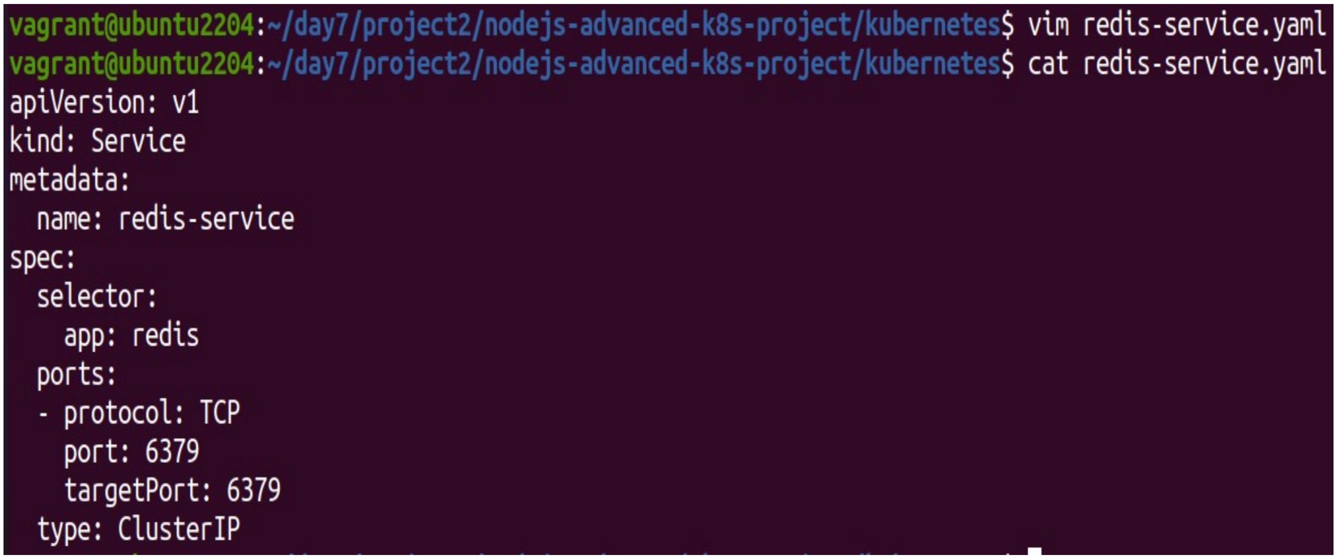


```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat redis-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    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
```

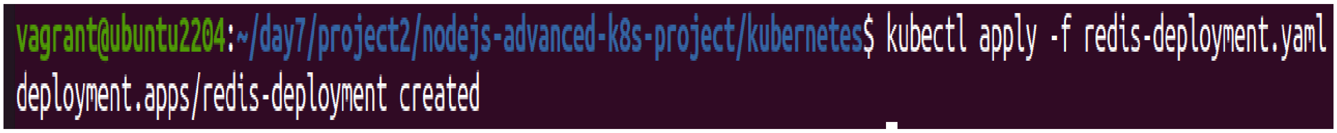


```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ vim redis-service.yaml
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ cat redis-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: redis-service
spec:
  selector:
    app: redis
  ports:
  - protocol: TCP
    port: 6379
    targetPort: 6379
  type: ClusterIP
```

2.8 Apply Kubernetes Configurations

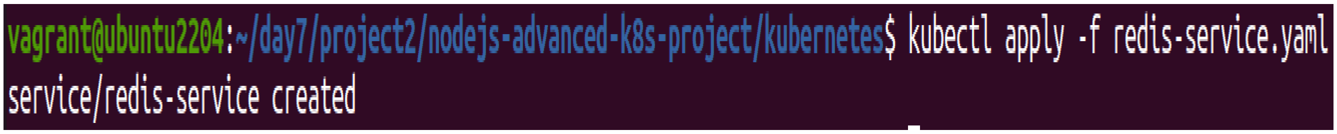
Apply all configurations to your Minikube cluster:

```
kubectl apply -f kubernetes/redis-deployment.yaml
```



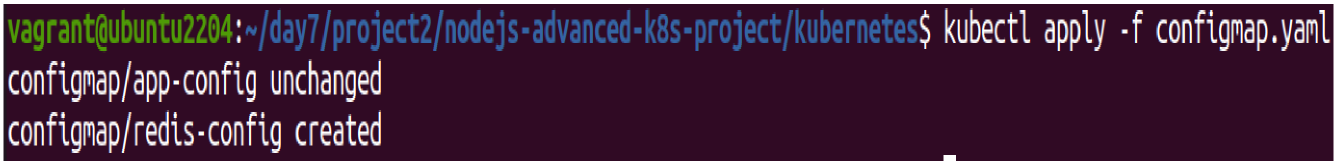
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f redis-deployment.yaml
deployment.apps/redis-deployment created
```

```
kubectl apply -f kubernetes/redis-service.yaml
```



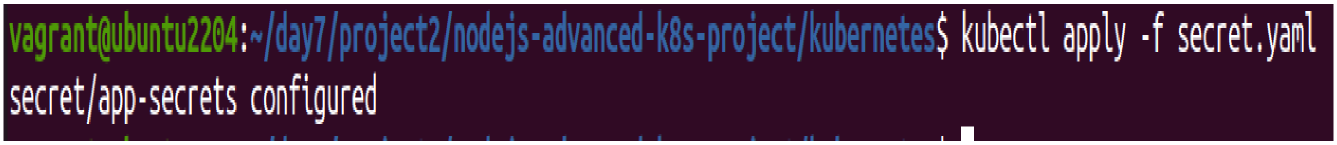
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f redis-service.yaml
service/redis-service created
```

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



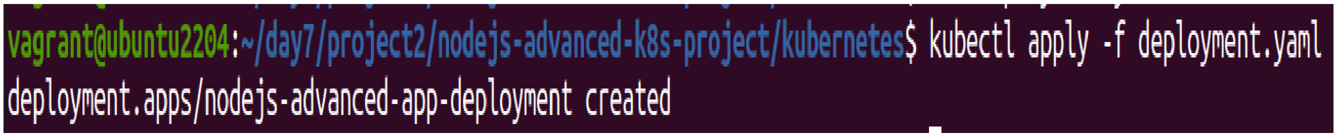
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f configmap.yaml
configmap/app-config unchanged
configmap/redis-config created
```

kubectl apply -f kubernetes/secret.yaml



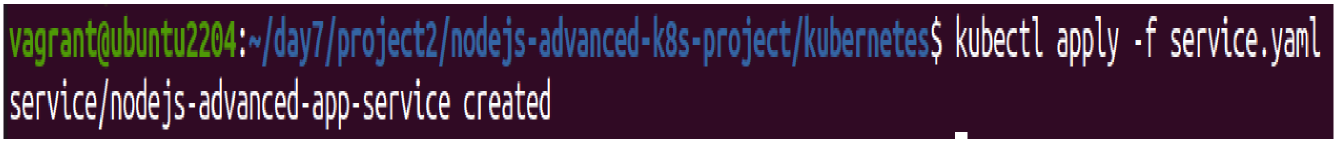
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f secret.yaml
secret/app-secrets configured
```

kubectl apply -f kubernetes/deployment.yaml



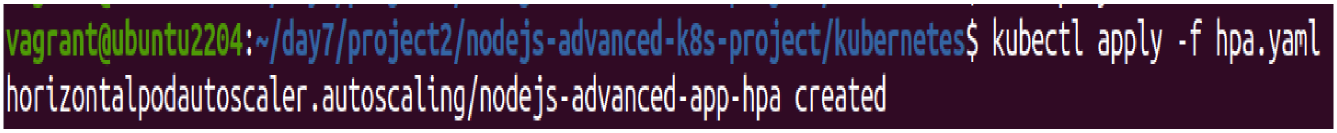
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-advanced-app-deployment created
```

kubectl apply -f kubernetes/service.yaml



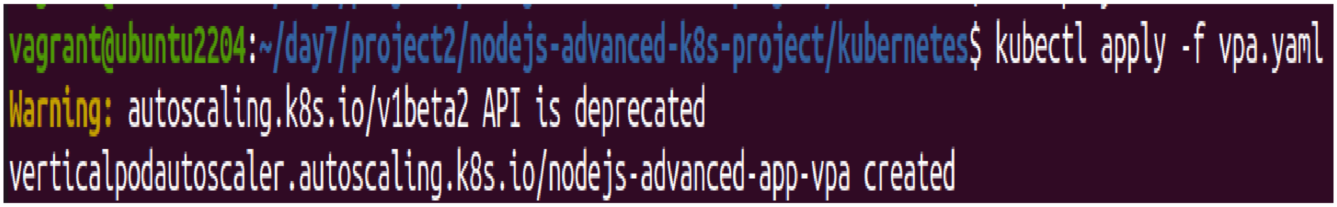
```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f service.yaml
service/nodejs-advanced-app-service created
```

kubectl apply -f kubernetes/hpa.yaml



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/nodejs-advanced-app-hpa created
```

kubectl apply -f kubernetes/vpa.yaml



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl apply -f vpa.yaml
Warning: autoscaling.k8s.io/v1beta2 API is deprecated
verticalpodautoscaler.autoscaling.k8s.io/nodejs-advanced-app-vpa created
```

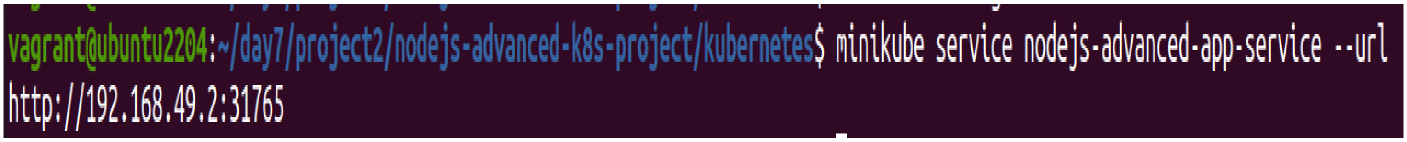
2.9 Verify Deployments and Services

Check the status of your deployments and services:

kubectl get all

Access the application via Minikube:

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



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ minikube service nodejs-advanced-app-service --url
http://192.168.49.2:31765
```

2.10 Testing Scaling

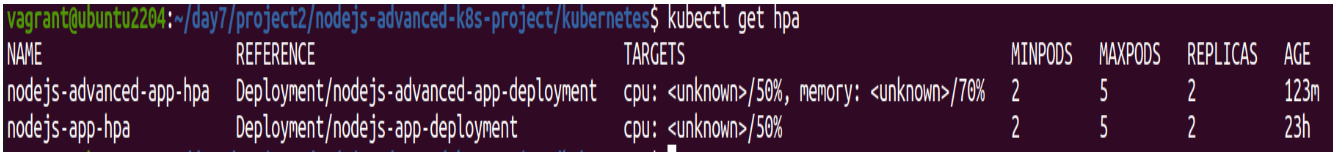
Simulate load on the application to test the HPA:

```
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
```



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ kubectl get hpa
NAME                    REFERENCE                                TARGETS                                      MINPODS   MAXPODS   REPLICAS   AGE
nodejs-advanced-app-hpa   Deployment/nodejs-advanced-app-deployment   cpu: <unknown>/50%, memory: <unknown>/70%   2         5         2          123m
nodejs-app-hpa          Deployment/nodejs-app-deployment            cpu: <unknown>/50%                           2         5         2          23h
```

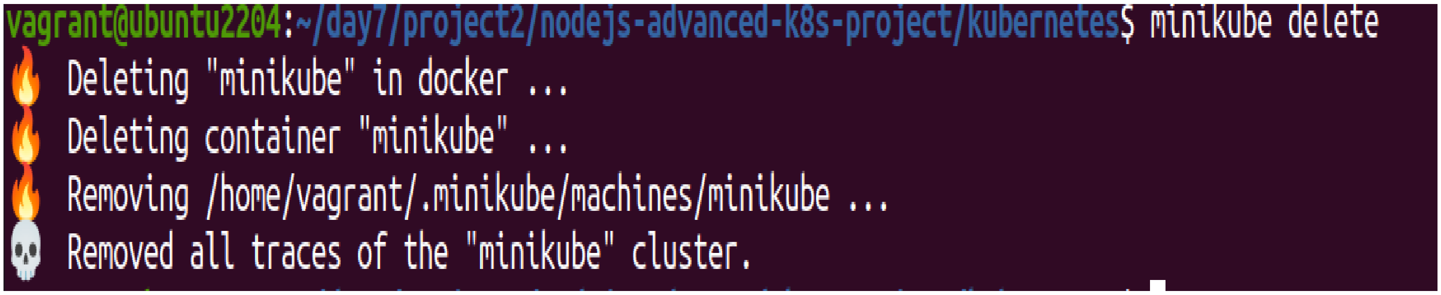
Watch the scaling events and verify that the application scales up and down based on the policies you configured.

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
```



```
vagrant@ubuntu2204:~/day7/project2/nodejs-advanced-k8s-project/kubernetes$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/vagrant/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
```