Project 01

Deploying a Node.js App Using Minikube Kubernetes

Overview

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

Prerequisites

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

Project Steps

1. Set Up Git Version Control

1.1. Initialize a Git Repository

Create a new directory for your project:

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

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

Initialize a Git repository:

```
git init
```

1.2. Create a Node.js Application

Initialize a Node.js project:

```
npm init -y
```

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7$ mkdir nodejs-k8s-project
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7$ cd nodejs-k8s-project/
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git init
Initialized empty Git repository in /home/einfochips/DevopsTraining/Assesment7/nodejs-k8s-project/.git/
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ npm init -y
Wrote to /home/einfochips/DevopsTraining/Assesment7/nodejs-k8s-project/package.json:

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

Install Express.js:

```
npm install express
```

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ npm install express

added 64 packages, and audited 65 packages in 5s

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

found 0 vulnerabilities
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ ls
node_modules  package.json  package-lock.json
```

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

```
const express = require('express');
const app = express();
const port = 3000;

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

app.listen(port, () => {
    console.log(`App running at http://localhost:${port}`);
});
```

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

   ```
   node_modules
   ```

1.3. Commit the Initial Code

Add files to Git:

```
git add .
```

Commit the changes:

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

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git add .
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ ls
index.js  node_modules  package.json  package-lock.json
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git status
On branch master

No commits yet

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

einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git commit -m "Initial commit with Node.js app"
[master (root-commit) 47ad59f] Initial commit with Node.js app
 4 files changed, 726 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 index.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
```

2. Branching and Fast-Forward Merge

2.1. Create a New Branch

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

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

2.2. Implement a New Route

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

```
app.get('/newroute', (req, res) => {
    res.send('This is a new route!');
});
```

Commit the changes:

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

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git checkout -b feature/add-route
Switched to a new branch 'feature/add-route'
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ vim index.js
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git commit -am "Add new route in index.js"
[feature/add-route 46145f1] Add new route in index.js
 1 file changed, 4 insertions(+)
```

2.3. Merge the Branch Using Fast-Forward

Switch back to the `main` branch:

```
git checkout main
```

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

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

Delete the feature branch:

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

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git checkout master
Switched to branch 'master'
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git merge feature/add-route
Updating 47ad59f..46145f1
Fast-forward
 index.js | 4 ++++
 1 file changed, 4 insertions(+)
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ git hist
* 46145f1 (HEAD -> master, feature/add-route) Add new route in index.js
* 47ad59f Initial commit with Node.js app
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$
```

3. Containerize the Node.js Application

3.1. Create a Dockerfile

Create a `Dockerfile` with the following content:

```
FROM node:14

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

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

3.2. Build and Test the Docker Image

Build the Docker image:

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

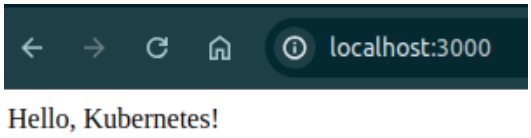
```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ vim Dockerfile
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ docker build -t nodejs-k8s-app .
[+] Building 13.5s (5/10)                                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                                  0.0s
 => => transferring dockerfile: 148B                                                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:14                                                                            3.1s
 => [auth] library/node:pull token for registry-1.docker.io                                                                           0.0s
 => [internal] load .dockerignore                                                                                                     0.0s
 => => transferring context: 2B                                                                                                       0.0s
 => [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa                     10.4s
 => => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa                      0.0s
 => => sha256:2cafa3fbb0b6529ee4726b4f599ec27ee557ea3dea7019182323b3779959927f 2.21kB / 2.21kB                                        0.0s
 => => sha256:1d12470fa662aa5cb58378dcdc8ea228c1745747db410bbefb8e2d0144b5453 7.51kB / 7.51kB                                         0.0s
 => => sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 1.05MB / 10.00MB                                       10.3s
 => => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 776B / 776B                                            0.0s
 => => sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 0B / 50.45MB                                           10.3s
 => => sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 1.05MB / 7.86MB                                        10.3s
 => [internal] load build context                                                                                                     0.3s
 => => transferring context: 2.24MB                                                                                                   0.2s
```

Run the Docker container to test:

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

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ docker run -p 3000:3000 nodejs-k8s-app
App running at http://localhost:3000
```

Access `http://localhost:3000` to see the app running.



Hello, Kubernetes!

4. Deploying to Minikube Kubernetes

4.1. Start Minikube

Start Minikube:

```
minikube start
```

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04 (vbox/amd64)
✨  Using the docker driver based on existing profile

❗  The requested memory allocation of 1963MiB does not leave room for system overhead (total system memory: 1963MiB). You may face stability issues.
💡  Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1963mb'

👍  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
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

4.2. Create Kubernetes Deployment and Service Manifests

Create a `deployment.yaml` file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app
spec:
  replicas: 2
  selector:
    matchLabels:
```

```
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - name: nodejs-app
        image: nodejs-k8s-app:latest
        ports:
        - containerPort: 3000
```

Create a `service.yaml` file for ClusterIP:

```
apiVersion: v1
kind: Service
metadata:
  name: nodejs-service
spec:
  selector:
    app: nodejs-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 3000
  type: ClusterIP
```

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

```
apiVersion: v1
kind: Service
metadata:
  name: nodejs-service-nodeport
spec:
  selector:
    app: nodejs-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 3000
    nodePort: 30001
  type: NodePort
```

```
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ vim deployment.yaml
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ vim service.yaml
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$ vim service-nodeport.yaml
einfochips@PUNELPT0436:~/DevopsTraining/Assesment7/nodejs-k8s-project$
```

4.3. Apply Manifests to Minikube

Apply the deployment:

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

Apply the ClusterIP service:

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

Apply the NodePort service:

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

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app created
vagrant@Master:~/Assesment7/nodejs-k8s-project$ kubectl apply -f service.yaml
service/nodejs-service created
vagrant@Master:~/Assesment7/nodejs-k8s-project$ kubectl apply -f service-nodeport.yaml
service/nodejs-service-nodeport created
vagrant@Master:~/Assesment7/nodejs-k8s-project$ kubectl get all
NAME                                  READY   STATUS             RESTARTS   AGE
pod/nodejs-app-867f6c98ff-47774       0/1     ImagePullBackOff   0          25s
pod/nodejs-app-867f6c98ff-gf2rz       0/1     ImagePullBackOff   0          25s

NAME                                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/kubernetes                    ClusterIP   10.96.0.1        <none>        443/TCP        22h
service/nodejs-service                ClusterIP   10.106.56.58     <none>        80/TCP         16s
service/nodejs-service-nodeport       NodePort    10.106.197.185   <none>        80:30001/TCP   5s

NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nodejs-app       0/2     2            0           25s

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/nodejs-app-867f6c98ff       2         2         0       25s
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

1. Access the application using the NodePort:

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

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ minikube ip
192.168.49.2
vagrant@Master:~/Assesment7/nodejs-k8s-project$
vagrant@Master:~/Assesment7/nodejs-k8s-project$ kubectl get all -o wide
NAME                          READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
pod/nodejs-app-8f885c978-h9vkp   1/1     Running   0          22m   10.244.0.25   minikube   <none>           <none>
pod/nodejs-app-8f885c978-rfv8d   1/1     Running   0          22m   10.244.0.26   minikube   <none>           <none>

NAME                           TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE   SELECTOR
service/kubernetes             ClusterIP   10.96.0.1        <none>        443/TCP        23h   <none>
service/nodejs-service         ClusterIP   10.107.228.168   <none>        80/TCP         89s   app=nodejs-app
service/nodejs-service-nodeport   NodePort    10.107.85.7      <none>        80:30001/TCP   78s   app=nodejs-app

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                              SELECTOR
deployment.apps/nodejs-app   2/2     2            2           22m   nodejs-app   chirag1212/my_repo:nodejs-k8s-app   app=nodejs-app

NAME                                DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                              SELECTOR
replicaset.apps/nodejs-app-8f885c978   2         2         2       22m   nodejs-app   chirag1212/my_repo:nodejs-k8s-app   app=nodejs-app,pod-template-ha
sh=8f885c978
vagrant@Master:~/Assesment7/nodejs-k8s-project$ curl http://192.168.49.2:30001
Hello, Kubernetes!vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

Making Changes to the App and Redeploying Using Kubernetes

6. Making Changes to the Node.js Application

6.1. Create a New Branch for Changes

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

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

6.2. Update the Application

Modify `index.js` to change the message:

```
const express = require('express');
const app = express();
const port = 3000;

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

app.get('/newroute', (req, res) => {
    res.send('This is a new route!');
});

app.listen(port, () => {
    console.log(`App running at http://localhost:${port}`);
});
```

6.3. Commit the Changes

Add and commit the changes:

```
git add .

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

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ curl http://192.168.49.2:30001
Hello, Kubernetes!vagrant@Master:~/Assesment7/nodejs-k8s-project$ git branch
  feature/add-route
* master
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git checkout -b feature/update-message
Switched to a new branch 'feature/update-message'
vagrant@Master:~/Assesment7/nodejs-k8s-project$ vim index.js
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git commit -am "Updated main route message"
[feature/update-message b84cb70] Updated main route message
 2 files changed, 2 insertions(+), 2 deletions(-)
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

7. Merge the Changes and Rebuild the Docker Image

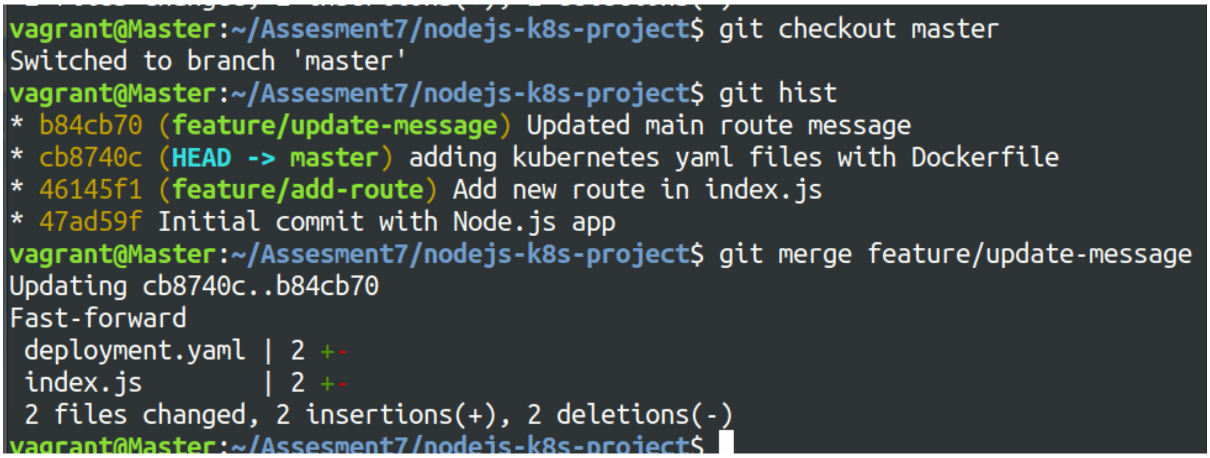
7.1. Merge the Feature Branch

Switch back to the `main` branch:

```
git checkout main
```

Merge the `feature/update-message` branch:

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



```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git checkout master
Switched to branch 'master'
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git hist
* b84cb70 (feature/update-message) Updated main route message
* cb8740c (HEAD -> master) adding kubernetes yaml files with Dockerfile
* 46145f1 (feature/add-route) Add new route in index.js
* 47ad59f Initial commit with Node.js app
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git merge feature/update-message
Updating cb8740c..b84cb70
Fast-forward
 deployment.yaml | 2 +-
 index.js        | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

Delete the feature branch:

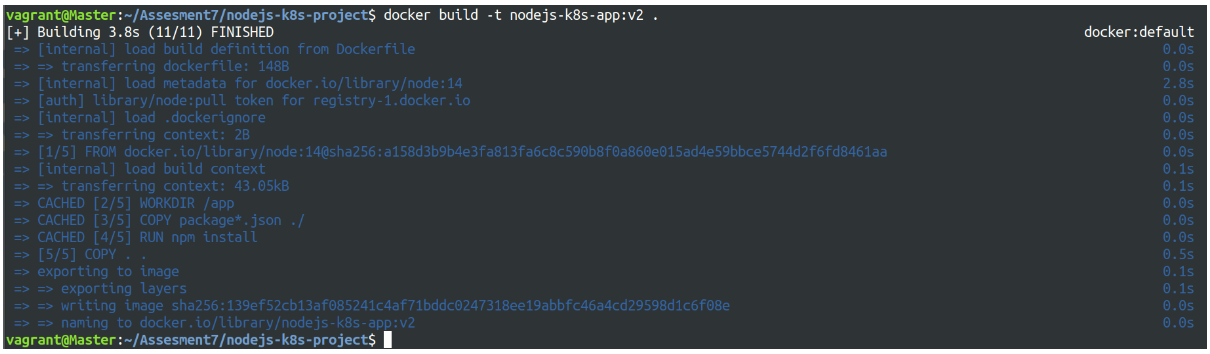
```
git branch -d feature/update-message
```

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git branch
  feature/add-route
  feature/update-message
* master
vagrant@Master:~/Assesment7/nodejs-k8s-project$ git branch -d feature/update-message
Deleted branch feature/update-message (was b84cb70).
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

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



```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ docker build -t nodejs-k8s-app:v2 .
[+] Building 3.8s (11/11) FINISHED                                                   docker:default
 => [internal] load build definition from Dockerfile                                          0.0s
 => => transferring dockerfile: 148B                                                          0.0s
 => [internal] load metadata for docker.io/library/node:14                                    2.8s
 => [auth] library/node:pull token for registry-1.docker.io                                   0.0s
 => [internal] load .dockerignore                                                             0.0s
 => => transferring context: 2B                                                               0.0s
 => [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa  0.0s
 => [internal] load build context                                                             0.1s
 => => transferring context: 43.05kB                                                          0.1s
 => CACHED [2/5] WORKDIR /app                                                                 0.0s
 => CACHED [3/5] COPY package*.json ./                                                        0.0s
 => CACHED [4/5] RUN npm install                                                              0.0s
 => [5/5] COPY . .                                                                            0.5s
 => exporting to image                                                                        0.1s
 => => exporting layers                                                                       0.1s
 => => writing image sha256:139ef52cb13af085241c4af71bddc0247318ee19abbfc46a4cd29598d1c6f08e  0.0s
 => => naming to docker.io/library/nodejs-k8s-app:v2                                          0.0s
vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

8. Update Kubernetes Deployment

8.1. Update the Deployment Manifest

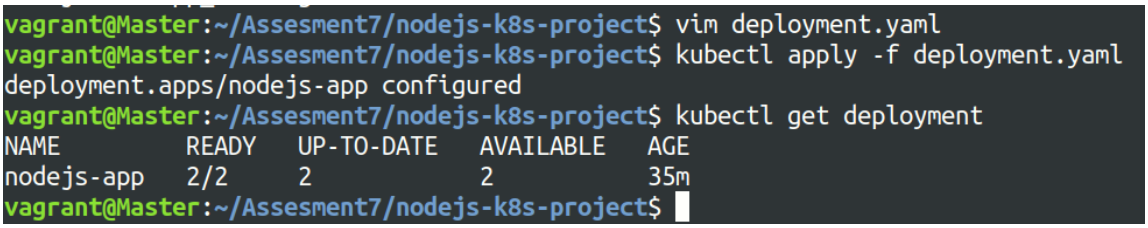
Modify `deployment.yaml` to use the new image version:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodejs-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nodejs-app
  template:
    metadata:
      labels:
        app: nodejs-app
    spec:
      containers:
      - name: nodejs-app
        image: nodejs-k8s-app:v2
        ports:
        - containerPort: 3000
```

8.2. Apply the Updated Manifest

Apply the updated deployment:

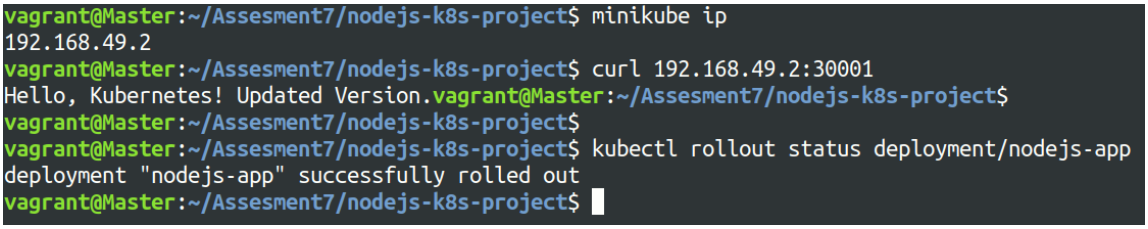
```
kubectl apply -f deployment.yaml
```



8.3. Verify the Update

Check the status of the deployment:

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



9. Access the Updated Application

9.1. Access Through ClusterIP Service

Forward the port to access the ClusterIP service:

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

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

9.2. Access Through NodePort Service

1. Access the application using the NodePort:

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

```
vagrant@Master:~/Assesment7/nodejs-k8s-project$ minikube ip
192.168.49.2
vagrant@Master:~/Assesment7/nodejs-k8s-project$ curl 192.168.49.2:30001
Hello, Kubernetes! Updated Version.vagrant@Master:~/Assesment7/nodejs-k8s-project$
```

Project 02

Deploying a Python Flask App Using Minikube Kubernetes

Overview

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

Prerequisites

- Minikube installed
- kubectl installed
- Git installed
- Python installed

Project Steps

1. Set Up Git Version Control

1.1. Initialize a Git Repository

Create a new directory for your project:

```
mkdir flask-k8s-project
```

```
cd flask-k8s-project
```

```
vagrant@Master:~/Assesment7$ ls
nodejs-k8s-project
vagrant@Master:~/Assesment7$ mkdir flask-k8s-project
vagrant@Master:~/Assesment7$ cd flask-k8s-project/
vagrant@Master:~/Assesment7/flask-k8s-project$ ls
```

Initialize a Git repository:
sh
Copy code
```
git init
```

```
vagrant@Master:~/Assesment7/flask-k8s-project$ git init
Initialized empty Git repository in /home/vagrant/Assesment7/flask-k8s-project/.git/
vagrant@Master:~/Assesment7/flask-k8s-project$
```

1.2. Create a Python Flask Application

Create a virtual environment:

```
python -m venv venv
```

```
source venv/bin/activate
```

```
vagrant@Master:~/Assesment7/flask-k8s-project$ python3 -m venv venv
vagrant@Master:~/Assesment7/flask-k8s-project$ source venv/bin/activate
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

Install Flask:
```
pip install Flask
```

```
vagrant@Master:~/Assesment7/flask-k8s-project$ source venv/bin/activate
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.7/101.7 KB 349.8 kB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.3-py3-none-any.whl (227 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 227.3/227.3 KB 351.6 kB/s eta 0:00:00
Collecting click>=8.1.3
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 97.9/97.9 KB 344.6 kB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.3/133.3 KB 381.3 kB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.3 blinker-1.8.2 click-8.1.7 flask-3.0.3 itsdangerous-2.2.0
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

Create an `app.py` file with the following content:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, Kubernetes!'
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Create a `requirements.txt` file to list the dependencies:
`Flask`

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

1.3. Commit the Initial Code

Add files to Git:

```
git add .
```

Commit the changes:

```
git commit -m "Initial commit with Flask app"
```

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim app.py
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim requirements.txt
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ ls
app.py  requirements.txt  venv
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim .gitignore
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        app.py
        requirements.txt

nothing added to commit but untracked files present (use "git add" to track)
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git add .
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git commit -m "Initial commit with Flask app"
[master (root-commit) aa64e55] Initial commit with Flask app
 3 files changed, 14 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 app.py
 create mode 100644 requirements.txt
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

2. Branching and Fast-Forward Merge

2.1. Create a New Branch

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

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

2.2. Implement a New Route

Modify `app.py` to add a new route:

```
@app.route('/newroute')

def new_route():

    return 'This is a new route!'
```

Commit the changes:

```
git add .

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim app.py
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ cat app.py

from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, Kubernetes!'

@app.route('/newroute')
def new_route():
    return 'This is a new route!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)


(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git commit -am "Added new route"
[feature/add-route 8ee98b5] Added new route
 1 file changed, 4 insertions(+)
```

2.3. Merge the Branch Using Fast-Forward

Switch back to the `main` branch:

```
git checkout main
```

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

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

Delete the feature branch:

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git checkout master
Switched to branch 'master'
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git merge feature/add-route
Updating aa64e55..8ee98b5
Fast-forward
 app.py | 4 ++++
 1 file changed, 4 insertions(+)
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git hist
* 8ee98b5 (HEAD -> master, feature/add-route) Added new route
* aa64e55 Initial commit with Flask app
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git branch -d feature/add-route
Deleted branch feature/add-route (was 8ee98b5).
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

3. Containerize the Flask Application

3.1. Create a Dockerfile

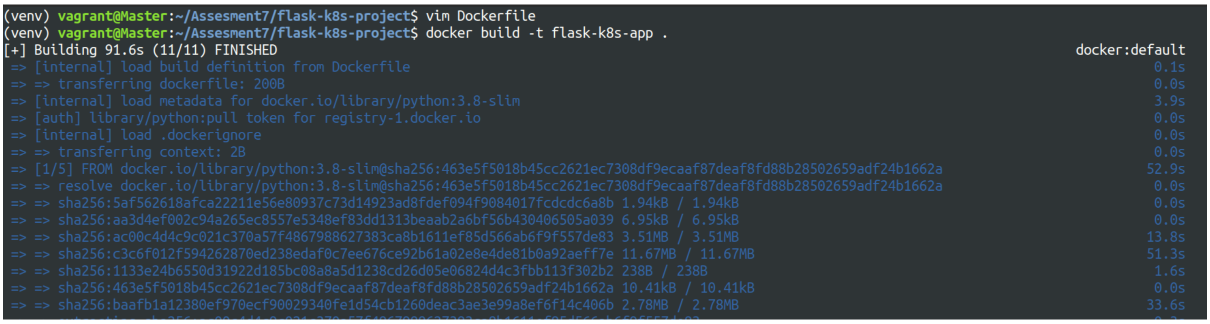
Create a `Dockerfile` with the following content:

```
FROM python:3.8-slim

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["python", "app.py"]
```

3.2. Build and Test the Docker Image

Build the Docker image:

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



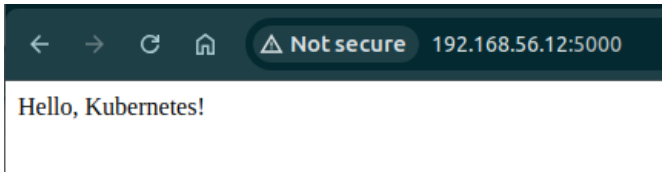
```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim Dockerfile
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ docker build -t flask-k8s-app .
[+] Building 91.6s (11/11) FINISHED                                                          docker:default
 => [internal] load build definition from Dockerfile                                                   0.1s
 => => transferring dockerfile: 200B                                                                   0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                                     3.9s
 => [auth] library/python:pull token for registry-1.docker.io                                          0.0s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [1/5] FROM docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28502659adf24b1662a  52.9s
 => => resolve docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28502659adf24b1662a  0.0s
 => => sha256:5af562618afca22211e56e80937c73d14923ad8fdef094f9084017fcdcdc6a8b 1.94kB / 1.94kB         0.0s
 => => sha256:aa3d4ef002c94a265ec8557e5348ef83dd1313beaab2a6bf56b430406505a039 6.95kB / 6.95kB         0.0s
 => => sha256:ac00c4d4c9c021c370a57f4867988627383ca8b1611ef85d566ab6f9f557de83 3.51MB / 3.51MB         13.8s
 => => sha256:c3c6f012f594262870ed238edaf0c7ee676ce92b61a02e8e4de81b0a92aeff7e 11.67MB / 11.67MB       51.3s
 => => sha256:1133e24b6550d31922d185bc08a8a5d1238cd26d05e06824d4c3fbb113f302b2 238B / 238B             1.6s
 => => sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28502659adf24b1662a 10.41kB / 10.41kB       0.0s
 => => sha256:baafb1a12380ef970ecf90029340fe1d54cb1260deac3ae3e99a8ef6f14c406b 2.78MB / 2.78MB         33.6s
```

Run the Docker container to test:

```
docker run -p 5000:5000 flask-k8s-app
```

1. Access `http://localhost:5000` to see the app running.

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ docker run -p 5000:5000 flask-k8s-app
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.3:5000
Press CTRL+C to quit
192.168.56.1 - - [16/Jul/2024 10:11:12] "GET / HTTP/1.1" 200 -
192.168.56.1 - - [16/Jul/2024 10:11:12] "GET /favicon.ico HTTP/1.1" 404 -
192.168.56.1 - - [16/Jul/2024 10:11:16] "GET / HTTP/1.1" 200 -
192.168.56.1 - - [16/Jul/2024 10:11:16] "GET / HTTP/1.1" 200 -
192.168.56.1 - - [16/Jul/2024 10:11:17] "GET / HTTP/1.1" 200 -
192.168.56.1 - - [16/Jul/2024 10:11:17] "GET / HTTP/1.1" 200 -
```



4. Deploying to Minikube Kubernetes

4.1. Start Minikube

Start Minikube:

```
minikube start
```

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ minikube start
😄  minikube v1.33.1 on Ubuntu 22.04 (vbox/amd64)
✨  Using the docker driver based on existing profile

❗  The requested memory allocation of 1963MiB does not leave room for system overhead (total system memory: 1963MiB). You may face stability issues.
💡  Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1963mb'

👍  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
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

🌟  Enabled addons: default-storageclass, storage-provisioner, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

4.2. Create Kubernetes Deployment and Service Manifests

Create a `deployment.yaml` file:

```
apiVersion: apps/v1

kind: Deployment

metadata:

  name: flask-app

spec:

  replicas: 2

  selector:
```

```
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
      - name: flask-app
        image: flask-k8s-app:latest
        ports:
        - containerPort: 5000
```

Create a `service.yaml` file for ClusterIP:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-service
spec:
  selector:
    app: flask-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 5000
  type: ClusterIP
```

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

```
apiVersion: v1
kind: Service
metadata:
  name: flask-service-nodeport
spec:
  selector:
    app: flask-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 5000
    nodePort: 30001
  type: NodePort
```

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim deployment.yaml
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim service.yaml
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim service-nodeport.yaml
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

4.3. Apply Manifests to Minikube

Apply the deployment:

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

Apply the ClusterIP service:

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

Apply the NodePort service:

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

```
Every 2.0s: kubectl get all -o wide                                                                 Master.localdomain: Tue Jul 16 10:22:13 2024

NAME                                READY   STATUS    RESTARTS        AGE     IP            NODE       NOMINATED NODE   READINESS GATES
pod/flask-app-66448dccdb-5qx98      1/1     Running   0               2m19s   10.244.0.43   minikube   <none>           <none>
pod/flask-app-66448dccdb-jbpxc      1/1     Running   0               47s     10.244.0.44   minikube   <none>           <none>
pod/nodejs-app-7fc4847d6c-5vbhb     1/1     Running   2 (9m45s ago)   3h17m   10.244.0.40   minikube   <none>           <none>
pod/nodejs-app-7fc4847d6c-bhkwh     1/1     Running   2               3h17m   10.244.0.36   minikube   <none>           <none>

NAME                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE     SELECTOR
service/kubernetes                  ClusterIP   10.96.0.1        <none>        443/TCP        27h     <none>
service/nodejs-service              ClusterIP   10.107.228.168   <none>        80/TCP         3h31m   app=nodejs-app
service/nodejs-service-nodeport     NodePort    10.107.85.7      <none>        80:30001/TCP   3h31m   app=nodejs-app

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES                                SELECTOR
deployment.apps/flask-app      2/2     2            2           3m23s   flask-app    chirag1212/my_repo:flask-k8s-app      app=flask-app
deployment.apps/nodejs-app     2/2     2            2           3h52m   nodejs-app   chirag1212/my_repo:nodejs-k8s-app_v2  app=nodejs-app

NAME                                      DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES                                SELECTOR
replicaset.apps/flask-app-66448dccdb      2         2         2       2m19s   flask-app    chirag1212/my_repo:flask-k8s-app      app=flask-app,pod-templa
te-hash=66448dccdb
replicaset.apps/flask-app-7b5c9d76c7      0         0         0       3m23s   flask-app    flask-k8s-app:latest                  app=flask-app,pod-templa
te-hash=7b5c9d76c7
replicaset.apps/nodejs-app-7fc4847d6c     2         2         2       3h17m   nodejs-app   chirag1212/my_repo:nodejs-k8s-app_v2  app=nodejs-app,pod-templ
ate-hash=7fc4847d6c
replicaset.apps/nodejs-app-8f885c978      0         0         0       3h52m   nodejs-app   chirag1212/my_repo:nodejs-k8s-app     app=nodejs-app,pod-templ
ate-hash=8f885c978
```

4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

Access the application using the NodePort:

```
curl http://<minikube-ip>:30002
```

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ curl 192.168.49.2:30002
Hello, Kubernetes!(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

5. Clean Up

Stop Minikube:

```
minikube stop
```

Delete Minikube cluster:

```
minikube delete
```

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/vagrant/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

6. Making Changes to the Flask Application

6.1. Create a New Branch for Changes

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

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

6.2. Update the Application

Modify `app.py` to change the message:

```
@app.route('/')
def hello_world():
    return 'Hello, Kubernetes! Updated version.'


@app.route('/newroute')
def new_route():
    return 'This is a new route!'
```

6.3. Commit the Changes

Add and commit the changes:

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ vim app.py
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git commit -am "Updated main route message"
[feature/update-message 199522f] Updated main route message
 1 file changed, 1 insertion(+), 1 deletion(-)
```

7. Merge the Changes and Rebuild the Docker Image

7.1. Merge the Feature Branch

Switch back to the `main` branch:

```
git checkout main
```

Merge the `feature/update-message` branch:

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

Delete the feature branch:

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git checkout master
Switched to branch 'master'
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git merge feature/update-message
Updating 8ee98b5..199522f
Fast-forward
 app.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ git branch -d feature/update-message
Deleted branch feature/update-message (was 199522f).
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ docker build -t flask-k8s-app:v2 .
[+] Building 5.6s (11/11) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 200B                                                                                         0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                                                           2.4s
 => [auth] library/python:pull token for registry-1.docker.io                                                                0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [1/5] FROM docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28502659adf24b1662a     0.0s
 => [internal] load build context                                                                                            0.3s
 => => transferring context: 185.12kB                                                                                        0.2s
 => CACHED [2/5] WORKDIR /app                                                                                                0.0s
 => CACHED [3/5] COPY requirements.txt requirements.txt                                                                      0.0s
 => CACHED [4/5] RUN pip install -r requirements.txt                                                                         0.0s
 => [5/5] COPY . .                                                                                                           2.5s
 => exporting to image                                                                                                       0.3s
 => => exporting layers                                                                                                      0.3s
 => => writing image sha256:16712ab1f865522836a2fbeaf0eb422efe624e5234d15e2bbd7c1dbcda1647ad                                 0.0s
 => => naming to docker.io/library/flask-k8s-app:v2                                                                          0.0s
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

8. Update Kubernetes Deployment

8.1. Update the Deployment Manifest

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

```
apiVersion: apps/v1

kind: Deployment

metadata:

  name: flask-app

spec:

  replicas: 2

  selector:

    matchLabels:

      app: flask-app
```

```
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
      - name: flask-app
        image: flask-k8s-app:v2
        ports:
        - containerPort: 5000
```

8.2. Apply the Updated Manifest

Apply the updated deployment:
sh
Copy code

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/flask-app created
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ kubectl get all
NAME                             READY   STATUS              RESTARTS   AGE
pod/flask-app-5bbdc7b87c-25hxz   0/1     ContainerCreating   0          4s
pod/flask-app-5bbdc7b87c-xq6fg   0/1     ContainerCreating   0          4s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   28s

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/flask-app   0/2     2            0           4s

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/flask-app-5bbdc7b87c   2         2         0       4s
```

8.3. Verify the Update

Check the status of the deployment:
sh
Copy code

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

```
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$ kubectl rollout status deployment/flask-app
deployment "flask-app" successfully rolled out
(venv) vagrant@Master:~/Assesment7/flask-k8s-project$
```

9. Access the Updated Application

9.1. Access Through ClusterIP Service

Forward the port to access the ClusterIP service:

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

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

9.2. Access Through NodePort Service

1. Access the application using the NodePort:

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