

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@PUNELPT0728:~/vagrant-vms/k8s/kubernetes-the-hard-way/vagrant$ cd
einfochips@PUNELPT0728:~$ cd DevOps-assessment/
einfochips@PUNELPT0728:~/DevOps-assessment$ cd Assessment\ 7/
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7$ ls
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7$ mkdir nodejs-k8s-project
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7$ cd nodejs-k8s-project/
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git init
Initialized empty Git repository in /home/einfochips/DevOps-assessment/Assessment 7/nodejs-k8s-project/.git/
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ npm init -y
Wrote to /home/einfochips/DevOps-assessment/Assessment 7/nodejs-k8s-project/package.json:

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

```

Install Express.js:

`npm install express`

```

einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ npm install express
added 64 packages, and audited 65 packages in 3s

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

found 0 vulnerabilities

```

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

```

const express = require('express');

const app = express();

const port = 3000;

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

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

```

1.

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

`node_modules`

```
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ code index.js
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ code .gitignore
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ ls
index.js  node_modules  package.json  package-lock.json
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ ls -la
total 56
drwxrwxr-x  4 einfochips einfochips 4096 Jul 16 10:08 .
drwxrwxr-x  3 einfochips einfochips 4096 Jul 16 10:05 ..
drwxrwxr-x  7 einfochips einfochips 4096 Jul 16 10:05 .git
-rw-rw-r--  1 einfochips einfochips  13 Jul 16 10:08 .gitignore
-rw-rw-r--  1 einfochips einfochips  237 Jul 16 10:06 index.js
drwxrwxr-x 66 einfochips einfochips 4096 Jul 16 10:06 node_modules
-rw-rw-r--  1 einfochips einfochips  282 Jul 16 10:06 package.json
-rw-rw-r--  1 einfochips einfochips 25511 Jul 16 10:06 package-lock.json
```

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@PUNELPT0728:~/DevOps-assessment/Assessment 7/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@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git commit -m "Initial commit with Node.js app"
[master (root-commit) 01cd984] Initial commit with Node.js app
 4 files changed, 724 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@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git checkout -b feature/add-route  
Switched to a new branch 'feature/add-route'  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ code index.js  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git add .  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git status  
On branch feature/add-route  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
        modified:   index.js  
  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git commit -m "Add new Route"  
[feature/add-route 3c04b49] Add new Route  
1 file changed, 4 insertions(+), 1 deletion(-)  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$
```

2.3. Merge the Branch Using Fast-Forward

Switch back to the `main` branch:

```
git checkout main
```

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

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

```
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git checkout main  
Switched to branch 'main'  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$ git merge --ff-only feature/add-route  
Already up to date.  
einfochips@PUNELPT0728:~/DevOps-assessment/Assessment 7/nodejs-k8s-project$
```

Delete the feature branch:

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

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:

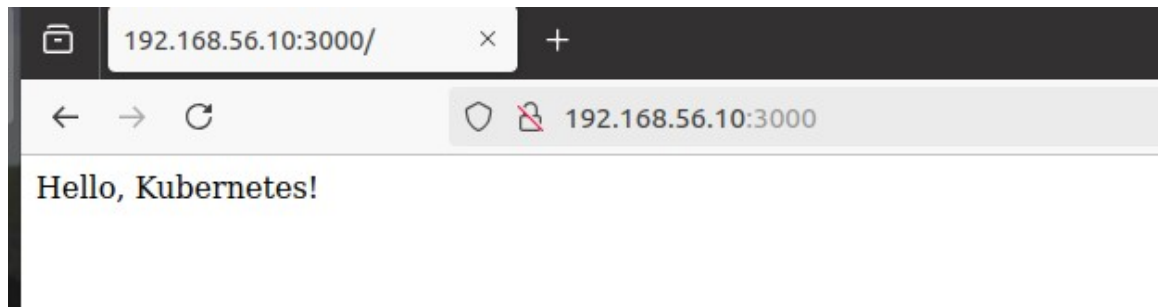
docker build -t nodejs-k8s-app .

```
[+] Building 1021.1s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 140B
=> [internal] load metadata for docker.io/library/node:14
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e50bbce5744d2f6fd8461aa
=> => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e50bbce5744d2f6fd8461aa
=> => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e50bbce5744d2f6fd8461aa 776B / 776B
=> => sha256:2c4f83f0b0b6529ee4726b5f599e27fe557ea34e70191823283779959927f 2.21kB / 2.21kB
=> => sha256:2ff1d7c41c74a25258bfa0f0bda0ba727f64518f55f65ca845ebc747976c4db 50.45MB / 50.45MB
=> => sha256:1d12470fa6e2a25cb50378ddc8ea28c173574db410bbefb8e2d9144b5452 7.51kB / 7.51kB
=> => sha256:b253aea7ea7e0671bb0008df01de101a38a045ff7bc656e3b0fbfc7c05ccas 7.86MB / 7.86MB
=> => sha256:3d2201bd995cccf12851a58820de3d34a17011dcb09ac9fd3a50c952cb131 10.69MB / 10.69MB
=> => sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9e2 51.89MB / 51.89MB
=> => sha256:d9abdf5894511ce28a05e2925a75e8a4acbd8634c39ad734fd8bae23d1b1569 191.85MB / 191.85MB
=> => extracting sha256:2ff1d7c41c74a25258bfa0f0bda0ba727f64518f55f65ca845ebc747976c4db
=> => sha256:d51ee850eaec899908e11bce4ebbf210eb1a31a0083f613aec0bcb903db 4.19kB / 4.19kB
=> => extracting sha256:b253aea7ea7e0671bb0008df01de101a38a045ff7bc656e3b0fbfc7c05ccas
=> => extracting sha256:3d2201bd995cccf12851a58820de3d34a17011dcb09ac9fd3a50c952cb131
=> => sha256:5f32ed3c3f278edd4fc671c880b5277355a29ae8f52b52cdf865f058378a590 35.24MB / 35.24MB
=> => extracting sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9e2
=> => sha256:0c8cc2f24a4dc04e02e086fc9440ba541e8acd9ad72d2e90df3ba22f158b3 2.29MB / 2.29MB
=> => sha256:0d27a8e801329087574c6760fba946d48e20d2c8e964e873de352603f22c4ceb 450B / 450B
=> => extracting sha256:d9abdf5894511ce28a05e2925a75e8a4acbd8634c39ad734fd8bae23d1b1569
=> => extracting sha256:d51ee850eaec899908e11bce4ebbf210eb1a31a0083f613aec0bcb903db
=> => extracting sha256:5f32ed3c3f278edd4fc671c880b5277355a29ae8f52b52cdf865f058378a590
=> => extracting sha256:0c8cc2f24a4dc04e02e086fc9440ba541e8acd9ad72d2e90df3ba22f158b3
=> => extracting sha256:0d27a8e801329087574c6760fba946d48e20d2c8e964e873de352603f22c4ceb
=> [internal] load build context
=> => transferring context: 2.24MB
=> [auth] library/node:pull token for registry-1.docker.io
=> [2/5] WORKDIR /app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> writing image sha256:043388221beb99dc757faa68820f5ddcdaaeabacadc5a1cc664931bb3dd9d9
=> naming to docker.io/library/nodejs-k8s-app
```

Run the Docker container to test:

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

1. Access <http://localhost:3000> to see the app running.



4. Deploying to Minikube Kubernetes

4.1. Start Minikube

Start Minikube:

```
minikube start
```

4.2. Create Kubernetes Deployment and Service Manifests

Create a `deployment.yaml` file:

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: nodejs-app
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
    matchLabels:
```

```
      app: nodejs-app
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: nodejs-app
```

```
    spec:
```

```
      containers:
```

```
- name: nodejs-app
  image: nodejs-k8s-app:latest
  ports:
    - containerPort: 3000
```

Create a `service.yaml` file for ClusterIP:

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

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

```
apiVersion: v1
kind: Service
metadata:
  name: nodejs-service-nodeport
spec:
  selector:
    app: nodejs-app
  ports:
    - protocol: TCP
      port: 80
```

targetPort: 3000

nodePort: 30001

type: NodePort

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@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ ls
deployment.yaml Dockerfile index.js node_modules nodesource_setup.sh package.json package-lock
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ nano deployment.yaml
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app created
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ kubectl apply -f service.yaml
service/nodejs-service created
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ kubectl apply -f service-nodeport.yaml
service/nodejs-service-nodeport created
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ watch kubectl get all -o wide
```

4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

1. Access the application using the NodePort:

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

```
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ minikube ip
192.168.49.2
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ curl http://192.168.49.2:30001
Hello, Kubernetes!vagrant@ubuntu2204:~/Assessment8-1/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@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ nano index.js
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ git add .
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ git commit -m "Update main route message"
[feature/update-message ad40280] Update main route message
5 files changed, 60 insertions(+), 1 deletion(-)
create mode 100644 Dockerfile
create mode 100644 deployment.yaml
create mode 100644 service-nodeport.yaml
create mode 100644 service.yaml

```

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
```

```

vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ git checkout main
Switched to branch 'main'
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ git merge --ff-only feature/update-message
Updating c571c6c..ad40280
Fast-forward
 Dockerfile      | 8 ++++++
 deployment.yaml | 19 ++++++
 index.js        | 7 +++++-
 service-nodeport.yaml | 14 ++++++
 service.yaml    | 13 ++++++
5 files changed, 60 insertions(+), 1 deletion(-)
create mode 100644 Dockerfile
create mode 100644 deployment.yaml
create mode 100644 service-nodeport.yaml
create mode 100644 service.yaml
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ git branch -d feature/update-message
Deleted branch feature/update-message (was ad40280).

```

7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

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

8. Update Kubernetes Deployment

8.1. Update the Deployment Manifest

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

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

8.2. Apply the Updated Manifest

Apply the updated deployment:

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

```
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nodejs-k8s-app       v2                 2a33e84394dd       11 seconds ago     919MB
daradesudarshan/centralrepo   latest            d271ed85b69a       2 hours ago        919MB
daradesudarshan/centralrepo   nodejs-k8s-app-v1 d271ed85b69a       2 hours ago        919MB
nodejs-k8s-app       latest            d271ed85b69a       2 hours ago        919MB
jenkins/jenkins      <none>            680da6c324fc       6 days ago         469MB
wordpress            latest            d2a2d7e671fd       3 weeks ago        685MB
nginx                latest            fffffc90d343       3 weeks ago        188MB
postgres             <none>            f23dc7cd74bd       2 months ago       432MB
gcr.io/k8s-minikube/kicbase   v0.0.44          5a6e59a9bdc0       2 months ago       1.26GB
mysql                5.7              5107333e08a8       7 months ago       501MB
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ docker image tag nodejs-k8s-app:v2 daradesudarshan/centralrepo:nodejs-k8s-app-v2
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ docker push daradesudarshan/centralrepo:nodejs-k8s-app-v2
The push refers to repository [docker.io/daradesudarshan/centralrepo]
01a5308eb478: Pushed
e0b48c3227b5: Layer already exists
584012320bb4: Layer already exists
9dca9a1ce4b3: Layer already exists
0d5f5a015e5d: Layer already exists
3c777d951de2: Layer already exists
f8a91dd5fc84: Layer already exists
cb81227abde5: Layer already exists
e01a454893a9: Layer already exists
c45660adde37: Layer already exists
fe0fb3ab4a0f: Layer already exists
f1186e5061f2: Layer already exists
b2dba7477754: Layer already exists
nodejs-k8s-app-v2: digest: sha256:ad1efc7b923a29e76e4bca73c3e77f1b6fc5cdcc8b8f7f402d145caebc7b139a size: 3050
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ nano deployment.yaml
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/nodejs-app configured
vagrant@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$
```

8.3. Verify the Update

Check the status of the deployment:

`kubectl rollout status deployment/nodejs-app`

```
vagrant@ubuntu2204:~$ kubectl rollout status deployment/nodejs-app
deployment "nodejs-app" successfully rolled out
vagrant@ubuntu2204:~$
```

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@ubuntu2204:~/Assessment8-1/nodejs-k8s-project$ curl http://192.168.49.2:30001
Hello, Kubernetes! Updated version.vagrant@ubuntu2204:~/Assessment8-1/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
```

Initialize a Git repository:

```
sh
```

Copy code

```
git init
```

1.2. Create a Python Flask Application

Create a virtual environment:

```
python -m venv venv
```

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

Install Flask:

```
sh
```

Copy code

```
pip install Flask
```

```

vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ python3 -m venv venv
vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ source venv/bin/activate
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ pip install Flask
Collecting Flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
    |#####| 101.7/101.7 KB 349.6 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 372.2 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 371.5 kB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.3-py3-none-any.whl (227 kB)
    |#####| 227.3/227.3 KB 365.0 kB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 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 Flask-3.0.3 Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.3 blinker-1.8.2 click-8.1.7 itsdangerous-2.2.0
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$

```

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

python

Copy code

```
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:

Copy code

```
Flask
```

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

Copy code

```
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@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano app.py
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano requirements.txt
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano .gitignore
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git add .
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git commit -m "initial commit with Flask app"
[main (root-commit) 18fd9a9] initial commit with Flask app
 3 files changed, 13 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 app.py
 create mode 100644 requirements.txt
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout -b feature/add-route
```

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@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout -b feature/add-route
Switched to a new branch 'feature/add-route'
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano app.py
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano app.py
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git add .
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git commit -m "Add new route"
[feature/add-route 21db6cd] Add new route
 1 file changed, 3 insertions(+)
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout main
Switched to branch 'main'
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```

2.3. Merge the Branch Using Fast-Forward

Switch back to the **main** branch:

```
git checkout main
```

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

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

Delete the feature branch:

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

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout main
Switched to branch 'main'
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git merge --ff-only feature/add-route
Updating 18fd9a9..21db6cd
Fast-forward
 app.py | 3 +++
 1 file changed, 3 insertions(+)
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git branch -d feature/add-route
Deleted branch feature/add-route (was 21db6cd).
(venv) vagrant@ubuntu2204:~/Assessment8-1/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:

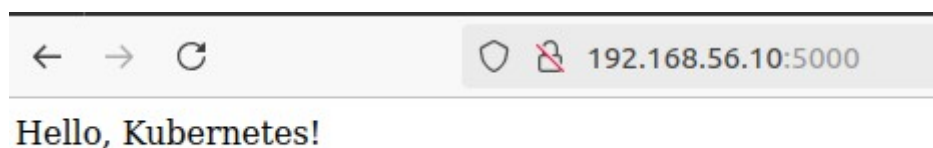
`docker build -t flask-k8s-app .`

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano Dockerfile
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ docker build -t flask-k8s-app .
[+] Building 85.8s (11/11) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 201B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim 3.4s
=> [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@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b2 52.0s
=> => resolve docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28 0.0s
=> => sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28502659adf24b1662a 10.41kB / 10.41kB 0.0s
=> => sha256:5af562618afca22211e56e80937c73d14923ad8fdef094f9084017fcdcdc6a8b 1.94kB / 1.94kB 0.0s
=> => sha256:aa3d4ef002c94a265ec8557e5348ef83dd1313beaab2a6bf56b430406505a039 6.95kB / 6.95kB 0.0s
=> => sha256:ac00c4d4c9c021c370a57f4867988627383ca8b1611ef85d566ab6f9f557de83 3.51MB / 3.51MB 21.6s
=> => sha256:c3c6f012f594262870ed238edaf0c7ee676ce92b61a02e8e4de81b0a92aeff7e 11.67MB / 11.67MB 51.2s
=> => sha256:1133e24b6550d31922d185bc08a8a5d1238cd26d05e06824d4c3fbb113f302b2 238B / 238B 1.8s
=> => sha256:baafb1a12380ef970ecf90029340fe1d54cb1260deac3ae3e99a8ef6f14c406b 2.78MB / 2.78MB 35.5s
=> => extracting sha256:ac00c4d4c9c021c370a57f4867988627383ca8b1611ef85d566ab6f9f557de83 0.1s
=> => extracting sha256:c3c6f012f594262870ed238edaf0c7ee676ce92b61a02e8e4de81b0a92aeff7e 0.5s
=> => extracting sha256:1133e24b6550d31922d185bc08a8a5d1238cd26d05e06824d4c3fbb113f302b2 0.0s
=> => extracting sha256:baafb1a12380ef970ecf90029340fe1d54cb1260deac3ae3e99a8ef6f14c406b 0.2s
=> [internal] load build context                                  0.4s
=> => transferring context: 20.02MB                                0.4s
=> [2/5] WORKDIR /app                                           0.1s
=> [3/5] COPY requirements.txt requirements.txt                 0.0s
=> [4/5] RUN pip install -r requirements.txt                    29.6s
=> [5/5] COPY . .                                               0.4s
=> exporting to image                                           0.2s
=> => exporting layers                                             0.2s
=> => writing image sha256:351529b10559e65649181b6ee5f65c2585c8cb8e9cdbc3867d51bd096aae24f0 0.0s
=> => naming to docker.io/library/flask-k8s-app                  0.0s
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ docker run -p 5000:5000 flask-k8s-app
```

Run the Docker container to test:

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

- 1.
2. Access <http://localhost:5000> to see the app running.



4. Deploying to Minikube Kubernetes

4.1. Start Minikube

Start Minikube:

`minikube start`

4.2. Create Kubernetes Deployment and Service Manifests

Create a `deployment.yaml` file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: 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
```

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
```

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano deployment.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl apply -f deployment.yaml
deployment.apps/flask-app created
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano service.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano service-nodeport.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl get deployment.yaml
error: the server doesn't have a resource type "deployment"
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl get deployment.apps
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
flask-app     0/2     2            0           86s
nodejs-app    2/2     2            2           75m
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl get deployment.apps -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES
flask-app     2/2     2            2           95s   flask-app    daradesudarshan/centralrepo:flask-k8s-app-v1
  app=flask-app
nodejs-app    2/2     2            2           76m   nodejs-app   daradesudarshan/centralrepo:nodejs-k8s-app_v2
  app=nodejs-app
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl apply -f service.yaml
service/flask-service created
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl apply -f service-nodeport.yaml
The Service "flask-service-nodeport" is invalid: spec.ports[0].nodePort: Invalid value: 30001: provided port is already allocated
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano service-nodeport.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl apply -f service-nodeport.yaml
service/flask-service-nodeport created
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano service-nodeport.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```

4.4. Access the Application

Get the Minikube IP:

```
minikube ip
```

Access the application using the NodePort:

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

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ curl http://192.168.49.2:30002
Hello, Kubernetes!(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```

5. Clean Up

Stop Minikube:

```
minikube stop
```

Delete Minikube cluster:

```
minikube delete
```

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@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout -b feature/update-message  
Switched to a new branch 'feature/update-message'  
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ nano app.py  
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git add .  
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git commit -m " Update main route message"  
[feature/update-message 1ba5552] Update main route message  
5 files changed, 59 insertions(+), 1 deletion(-)  
create mode 100644 Dockerfile  
create mode 100644 deployment.yaml  
create mode 100644 service-nodeport.yaml  
create mode 100644 service.yaml  
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-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
```

1.

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@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git checkout main
Switched to branch 'main'
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git merge --ff-only feature/update-message
Updating 21db6cd..1ba5552
Fast-forward
 Dockerfile      | 14 ++++++++
 app.py          |  2 +-
 deployment.yaml | 19 ++++++++
 service-nodeport.yaml | 13 ++++++++
 service.yaml    | 12 ++++++++
 5 files changed, 59 insertions(+), 1 deletion(-)
 create mode 100644 Dockerfile
 create mode 100644 deployment.yaml
 create mode 100644 service-nodeport.yaml
 create mode 100644 service.yaml
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ git branch -d feature/update-message
Deleted branch feature/update-message (was 1ba5552).
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```

7.2. Rebuild the Docker Image

Rebuild the Docker image with a new tag:

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

```

start a build
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ docker build -t flask-k8s-app:v2 .
[+] Building 3.0s (11/11) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 201B                             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@sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8fd88b28 0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 186.94kB                             0.0s
=> 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 . .                                              0.3s
=> exporting to image                                           0.2s
=> => exporting layers                                           0.2s
=> => writing image sha256:3fc4761e40a9336196399ca051f01ffa77ea1696ac76d8822f8e241ab81c0fc0 0.0s
=> => naming to docker.io/library/flask-k8s-app:v2             0.0s
(venv) vagrant@ubuntu2204:~/Assessment8-1/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@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl get deployment -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                                     SELECTOR
flask-app     2/2     2            2           13m   flask-app    daradesudarshan/centralrepo:flask-k8s-app-v2   app=flask-app
nodejs-app    2/2     2            2           87m   nodejs-app   daradesudarshan/centralrepo:nodejs-k8s-app_v2   app=nodejs-app
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```

8.3. Verify the Update

Check the status of the deployment:

sh

Copy code

`kubectl rollout status deployment/flask-app`

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl rollout status deployment/flask-app
deployment "flask-app" successfully rolled out
(venv) vagrant@ubuntu2204:~/Assessment8-1/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`

```
(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ kubectl port-forward service/flask-service 8080:80
Forwarding from 127.0.0.1:8080 -> 5000
^C(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$ curl http://192.168.49.2:30002
Hello, Kubernetes! Updated version(venv) vagrant@ubuntu2204:~/Assessment8-1/flask-k8s-project$
```