□ Step-by-Step: Creating a Web Application using Kubernetes

🔹 Step 1: Install Required Software

Software	Purpose	Download/Command
1. Docker	To build and run container images	📎 https://www.docker.com/products/docker-desktop
2. kubectl	CLI tool to interact with Kubernetes clusters	📎 https://kubernetes.io/docs/tasks/tools/
3. Minikube	Local Kubernetes cluster for testing	📎 https://minikube.sigs.k8s.io/docs/start/
4. VS Code / IDE	To write and manage code	📎 https://code.visualstudio.com/
5. Web Browser	To access your running web app	Chrome / Edge / Firefox

---

🔹 Step 2: Verify Installation

Open your terminal (or PowerShell on Windows) and check:

docker --version

kubectl version --client

minikube version

🔹 Step 3: Start a Local Kubernetes Cluster

minikube start

✅ This command creates a local Kubernetes cluster using Docker as the driver.

Check cluster status:

kubectl get nodes

You should see one node named something like minikube.

🔹 Step 4: Create a Simple Web Application

Example: Python Flask App (app.py)

```
from flask import Flask

app = Flask(__name__)


@app.route('/')

def home():

   return "Hello from Kubernetes!"


if __name__ == '__main__':

   app.run(host='0.0.0.0', port=5000)
```

Create a requirements.txt

```
flask
```

🔷 Step 5: Create a Dockerfile

```
FROM python:3.9-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

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

🔷 Step 6: Build and Run Docker Image

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

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

✅ Open browser → http://localhost:5000 → You should see
"Hello from Kubernetes!"

🔷 Step 7: Create Kubernetes Deployment and Service Files

📄 deployment.yaml

```
apiVersion: apps/v1

kind: Deployment

metadata:

  name: flask-deployment
```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
      - name: flask-container
        image: flask-k8s-app
        ports:
        - containerPort: 5000
```

📄 **service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: flask-service
spec:
  type: NodePort
  selector:
    app: flask-app
  ports:
    - port: 5000
      targetPort: 5000
      nodePort: 30007
```

🔷 **Step 8: Apply Deployment and Service**

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

kubectl apply -f service.yaml

Check status:

kubectl get pods

kubectl get svc

◈ Step 9: Access the Web App

Start Minikube tunnel or get the URL:

minikube service flask-service

✅ It will open your web app in the browser — running inside Kubernetes!

◈ Step 10: Manage Your Cluster

Command	Description
kubectl get pods	Lists running pods
kubectl logs <pod-name>	Shows app logs
kubectl delete -f deployment.yaml	Deletes deployment
minikube stop	Stops cluster
minikube delete	Removes cluster

□ Optional (For Production)

If deploying to a cloud platform:

- Use **Google Kubernetes Engine (GKE)**, **AWS EKS**, or **Azure AKS**
- Push your image to **Docker Hub** or **Google Container Registry**
- Update the image path in your deployment YAML file.