✅ Environment Setup Summary: Host a Webpage on Minikube

💻 1. Local Setup (Prerequisites)

- **OS:** Windows 11
- **Tools Installed:**
  - Docker Desktop
  - Minikube
  - kubectl CLI
  - Hyper-V (used as the Minikube driver)

🧱 2. Create Project Files

Folder Path: C:\Damith\k8s-demo
Inside it, add:

index.html

Your simple HTML page with a link to a CSS file.

style.css

Your custom CSS file to style the webpage.

Dockerfile

Dockerfile

CopyEdit

FROM nginx:alpine

COPY index.html /usr/share/nginx/html/index.html

COPY style.css /usr/share/nginx/html/style.css

Description:

- Uses a lightweight nginx:alpine image.
- Copies HTML and CSS to Nginx's default root folder so they are served on container start.

🐬 3. Build Docker Image

Before building, set Docker environment to Minikube:

bash

CopyEdit

@FOR /f "tokens=*" %i IN ('minikube docker-env') DO @%i

Then build your image:

bash

CopyEdit

docker build -t damith1 C:\Damith\k8s-demo

- -t damith1 gives your image the name/tag **damith1**.
- This image is stored in Minikube's internal Docker registry.

🚀 4. Create Kubernetes Deployment

Create a file named deployment.yaml:

yaml

CopyEdit

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: damith1-deployment
spec:
 replicas: 1
 selector:
  matchLabels:
   app: damith1
 template:
  metadata:
   labels:
    app: damith1
  spec:
   containers:
   - name: damith1
    image: damith1
    imagePullPolicy: Never
    ports:
    - containerPort: 80
```

Description:

- Creates a Kubernetes **Deployment** running one pod with the damith1 Docker image.
- imagePullPolicy: Never tells Kubernetes to use the local image instead of pulling from a registry.

Apply the deployment:

bash

CopyEdit

kubectl apply -f deployment.yaml

🌐 5. Expose the Deployment as a Service

Run:

bash

CopyEdit

kubectl expose deployment damith1-deployment --type=NodePort --port=80

Description:

- Exposes your pod to the outside world using a **NodePort**.

🔗 6. Access the Webpage

Get the service URL:

bash

CopyEdit

minikube service damith1-deployment --url

Open the URL in your browser — you'll see your styled demo webpage!