Manage Kubernetes Resources Using CLI

Goal: Learn to install the tools and manage basic Kubernetes resources using the `kubectl` CLI: create pods and deployments, scale, expose as a service.

Prerequisites

- Laptop/desktop with **at least 4 GB RAM** and internet access.
- One of these OS options: **Ubuntu/Debian Linux**, **macOS**, or **Windows 10/11**.
- Basic terminal or PowerShell familiarity.

Quick Checklist Before You Start

- Terminal (Linux/macOS) or PowerShell (Windows) ready.
- Admin/sudo access to install software.
- Docker and Minikube installed.

Install the Tools

We will use **Minikube** for the local Kubernetes cluster. It runs a single-node cluster inside Docker.

You need: **Docker (or Docker Desktop)**, **kubectl**, and **Minikube**.

A. Install Docker

Windows:

1. Download and install **Docker Desktop** from Docker's website.
2. Enable **WSL2 backend** during installation.
3. Start Docker Desktop and ensure it is running.

Verify Docker installation:

```
docker --version
```

B. Install kubectl (Kubernetes CLI)

Windows (using Chocolatey):

```
choco install kubernetes-cli
```

Verify installation:

```
kubectl version --client
```

C. Install and Start Minikube

Windows (using Chocolatey):

```
choco install minikube
minikube start --driver=docker
```

Verify the cluster is running:

```
minikube status
kubectl get nodes
```

If `kubectl get nodes` returns a node in **Ready** state — you are ready to go!

�� Step 1: Create a Simple

Pod **Create a file:** `nginx-pod.yaml`

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:latest
    ports:
    - containerPort: 80
```

Apply the manifest:

```
kubectl apply -f nginx-pod.yaml
```

Check pod status:

```
kubectl get pods
# expected: nginx-pod 1/1 Running
```

Describe the pod (to see detailed info):

```
kubectl describe pod nginx-pod
```

Forward the port locally to access it in the browser:

```
kubectl port-forward pod/nginx-pod 8080:80
```

Then open http://localhost:8080 in your browser.

When done, delete the pod:

```
kubectl delete pod nginx-pod
```

�� Step 2: Deployment and Scaling

Create a deployment:

```
kubectl create deployment my-nginx --image=nginx
```

Check deployments and pods:

```
kubectl get deployments
kubectl get pods -l app=my-nginx
```

Scale to 3 replicas:

```
kubectl scale deployment my-nginx --replicas=3
kubectl get pods
```

Update the image (rolling update):

```
kubectl set image deployment/my-nginx nginx=nginx:1.25
kubectl rollout status deployment/my-nginx
```

If something goes wrong, rollback:

```
kubectl rollout undo deployment/my-nginx
```

�� Step 3: Expose Deployment as a Service

Expose the deployment using a NodePort service:

```
kubectl expose deployment my-nginx --type=NodePort --port=80
kubectl get svc
```

Get the URL of your app using Minikube:

```
minikube service my-nginx --url
```

Open the displayed URL in your browser.
To remove the service:

```
kubectl delete svc my-nginx
```

�� Step 4: Cleanup

After the lab, clean up all created resources:

```
kubectl delete deployment my-nginx
kubectl delete svc my-nginx
kubectl delete all --all -n default
```

If you want to stop or delete the cluster:

```
minikube stop
minikube delete
```

�� Summary

In this lab, you learned how to:

1. Install Docker, kubectl, and Minikube.

2. Create and manage Pods.

3. Deploy and scale applications. 4.

Expose Deployments as services. 5.

Clean up resources safely.