Setting Up a Basic Kubernetes Cluster Using Docker Desktop

Docker Desktop provides an easy way to set up and manage a Kubernetes cluster locally. Here's a step-by-step guide:

---

1. Prerequisites

Before starting, ensure the following are in place:

- **Docker Desktop Installed:**
    - Download and install Docker Desktop from the Docker official website.
- **System Requirements:**
    - Windows 10/11 (Pro, Enterprise, or Education) or macOS 10.14+.
    - At least 4GB of RAM and a modern processor.
- **Enable WSL 2 (Windows only):**
    - Install and configure WSL 2 for Docker Desktop if you're on Windows.

---

2. Enable Kubernetes in Docker Desktop

1. **Open Docker Desktop:** Launch Docker Desktop from your applications menu.
2. **Access Settings:**
    - On the Docker Desktop dashboard, click on the **gear icon** to open settings.
3. **Enable Kubernetes:**
    - Navigate to the **Kubernetes** tab.
    - Check the box for **Enable Kubernetes**.
    - Click **Apply & Restart** to enable Kubernetes. Docker Desktop will configure a local Kubernetes cluster.
4. **Verify the Setup:**
    - Open a terminal and type:

bash

kubectl version --client

kubectl cluster-info

3. Deploy a Simple Application

1. **Create a Deployment:**
    - Write a simple deployment YAML file (e.g., nginx-deployment.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.21
        ports:
        - containerPort: 80
```

Apply the Deployment:

- Save the YAML file and run:

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

Check the Pods:

- Verify the deployment and running pods:

```
Bash
kubectl get pods
```

4. Expose the Application

1. **Create a Service:**
    - Expose the deployment using a NodePort:

```
bash
```

kubectl expose deployment nginx-deployment --type=NodePort --name=nginx-service

Get the Service Details:

- Find the port to access the service:

bash

kubectl get svc

Access the Application:

- Open a browser and go to:

http://localhost:<NodePort>

5. Manage the Cluster

- **Scale the Deployment:**

bash

kubectl scale deployment nginx-deployment --replicas=4

Verify the scaling:

bash

kubectl get pods

Delete Resources:

- To clean up:

Bash

kubectl delete svc nginx-service

kubectl delete deployment nginx-deployment