Setting Up a Basic Kubernetes Cluster Using Docker

1.Prerequisites Before starting, ensure the following are in place:

• Docker Desktop Installed: o Download and install Docker Desktop from the Docker official website.

• System Requirements: o Windows 10/11 (Pro, Enterprise, or Education) or macOS 10.14+. o At least 4GB of RAM and a modern processor.

• Enable WSL 2 (Windows only): o Install and configure WSL 2 for Docker Desktop if you're on Windows.

2. Enable Kubernetes in Docker Desktop:

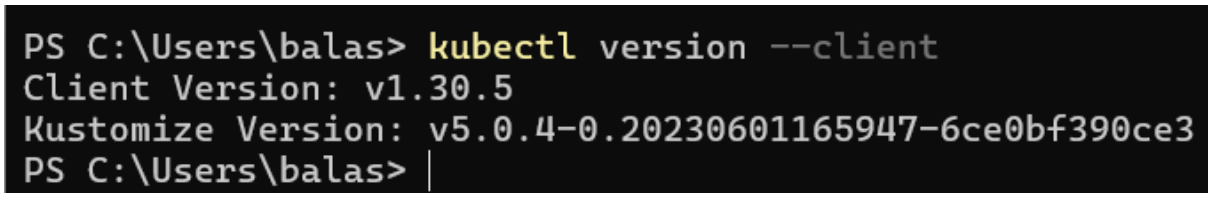
1. Open Docker Desktop: Launch Docker Desktop from your applications menu.

2. Access Settings: o On the Docker Desktop dashboard, click on the gear icon to open settings.

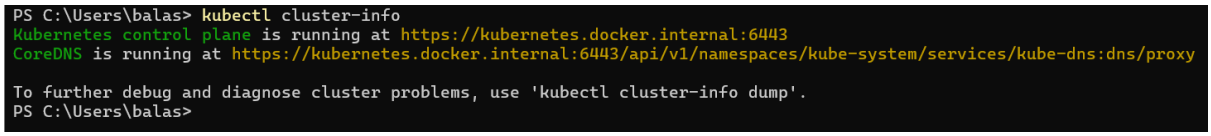
3. Enable Kubernetes: o Navigate to the Kubernetes tab. o Check the box for Enable Kubernetes. o Click Apply & Restart to enable Kubernetes. Docker Desktop will configure a local Kubernetes cluster.

4. Verify the Setup: o Open a terminal and type:

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)

docker run -d -p 8080:8080 -p 50000:50000 --name jenkins -v jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock jenkins/jenkins:lts

Apply the Deployment:

• Save the YAML file and run:

kubectl apply -f nginx-deployment.yaml

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment unchanged
PS C:\Users\balas\OneDrive\Desktop\Kubernetes>
```

Check the Pods:

• Verify the deployment and running pods:

kubectl get pods

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl get pods
NAME                                READY   STATUS    RESTARTS        AGE
nginx-deployment-74676ff58f-6zs4w   1/1     Running   1 (3m43s ago)   19h
nginx-deployment-74676ff58f-c5rjq   1/1     Running   1 (3m43s ago)   19h
```

4. Expose the Application:

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

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

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl expose deployment nginx-deployment --type=NodePort --name=nginx-s
ervice
service/nginx-service exposed
PS C:\Users\balas\OneDrive\Desktop\Kubernetes>
```

Get the Service Details: Find the port to access the service:

kubectl get svc

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl get svc
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP        20h
nginx-service   NodePort    10.108.99.112   <none>        80:32036/TCP   22s
```

Access the Application:

• Open a browser and go to: http://localhost:<NodePort> (Here : localhost:32036)

5. Manage the Cluster

• Scale the Deployment:

kubectl scale deployment nginx-deployment --replicas=4

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl scale deployment nginx-deployment --replicas=4
deployment.apps/nginx-deployment scaled
```

Verify the scaling:

kubectl get pods

```
deployment.apps/nginx-deployment scaled
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-74676ff58f-gqj6l   1/1     Running   0          19s
nginx-deployment-74676ff58f-st8rg   1/1     Running   0          2m58s
nginx-deployment-74676ff58f-x874t   1/1     Running   0          19s
nginx-deployment-74676ff58f-xq9bp   1/1     Running   0          2m58s
PS C:\Users\balas\OneDrive\Desktop\Kubernetes>
```

Delete Resources:

To clean up:

kubectl delete svc nginx-service

kubectl delete deployment nginx-deployment

```
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl delete svc nginx-service
service "nginx-service" deleted
PS C:\Users\balas\OneDrive\Desktop\Kubernetes> kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
PS C:\Users\balas\OneDrive\Desktop\Kubernetes>
```

Check the page created again