Practical-9

Performing basics commands to interact with Kubernetes.

The objective of this lab is to familiarize yourself with basic commands to interact with a Kubernetes cluster. You will learn how to perform essential operations such as deploying pods,

Lab Steps:

Step 1: Verify `kubectl` Configuration

```
PS D:\Desktop\stream> kubectl config current-context
docker-desktop
```

Step 2: List Nodes To view the nodes in your Kubernetes cluster, use the following command

```
PS D:\Desktop\stream> kubectl get nodes
NAME             STATUS   ROLES           AGE     VERSION
docker-desktop   Ready    control-plane   2m30s   v1.27.2
```

Step 3: Create a Deployment Create a simple NGINX deployment using the `kubectl create` command

```
PS D:\Desktop\stream> kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
```

Verify the deployment

```
PS D:\Desktop\stream> kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           27s
```

Step 4: List Pods To list the pods in your cluster, run

```
PS D:\Desktop\stream> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-66fb7f764c-8jv7g   1/1     Running   0          3m9s
```

Step 5: Access Pod Logs Access the logs of one of the NGINX pods to check its activity

```
PS D:\Desktop\stream> kubectl logs nginx-deployment-66fb7f764c-8jv7g
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/11/23 13:05:43 [notice] 1#1: using the "epoll" event method
2023/11/23 13:05:43 [notice] 1#1: nginx/1.25.3
2023/11/23 13:05:43 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/11/23 13:05:43 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/11/23 13:05:43 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/11/23 13:05:43 [notice] 1#1: start worker processes
2023/11/23 13:05:43 [notice] 1#1: start worker process 29
2023/11/23 13:05:43 [notice] 1#1: start worker process 30
2023/11/23 13:05:43 [notice] 1#1: start worker process 31
2023/11/23 13:05:43 [notice] 1#1: start worker process 32
2023/11/23 13:05:43 [notice] 1#1: start worker process 33
2023/11/23 13:05:43 [notice] 1#1: start worker process 34
2023/11/23 13:05:43 [notice] 1#1: start worker process 35
2023/11/23 13:05:43 [notice] 1#1: start worker process 36
```

Step 6: Expose Deployment as a Service Expose the NGINX deployment as a service to make it accessible externally

```
PS D:\Desktop\stream> kubectl expose deployment nginx-deployment --port=80 --type=NodePort --name=nginx-service
service/nginx-service exposed
```

Step 7: List Services To list the services in your cluster

```
PS D:\Desktop\stream> kubectl get services
NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes         ClusterIP   10.96.0.1       <none>        443/TCP        10m
nginx-deployment   ClusterIP   10.96.11.106    <none>        80/TCP         2m31s
nginx-service      NodePort    10.99.103.113   <none>        80:32031/TCP   91s
```

Step 8: Access the NGINX Service Determine the NodePort assigned to the NGINX service

```
PS D:\Desktop\stream> kubectl describe service nginx-deployment
Name:              nginx-deployment
Namespace:         default
Labels:            app=nginx-deployment
Annotations:       <none>
Selector:          app=nginx-deployment
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.11.106
IPs:               10.96.11.106
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         10.1.0.6:80
Session Affinity:  None
Events:            <none>
```

Step 9: Delete Resources Clean up by deleting the deployment and service

```
PS D:\Desktop\stream> kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
PS D:\Desktop\stream> kubectl delete service nginx-deployment
service "nginx-deployment" deleted
PS D:\Desktop\stream>
```

Step 10: Scale Deployment Scale the NGINX deployment to run multiple replicas

```
PS D:\Desktop\stream> kubectl scale deployment nginx-deployment --replicas=3
error: no objects passed to scale
```

Step 11: Update Deployment

```
PS D:\Desktop\stream> kubectl create deployment nginx-deployment --image=nginx:1.21
deployment.apps/nginx-deployment created
PS D:\Desktop\stream> kubectl scale deployment nginx-deployment --replicas=3
deployment.apps/nginx-deployment scaled
PS D:\Desktop\stream> kubectl set image deployment/nginx-deployment nginx=nginx:1.21
PS D:\Desktop\stream> kubectl set image deployment/nginx-deployment nginx=nginx:1.21
```

Verify the rollout status :

```
PS D:\Desktop\stream> kubectl rollout status deployment/nginx-deployment
deployment "nginx-deployment" successfully rolled out
```

Step 12: Rollback Deployment If needed, you can rollback to the previous deployment version

```
PS D:\Desktop\stream> kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>

PS D:\Desktop\stream> kubectl set image deployment/nginx-deployment nginx=nginx:1.22
deployment.apps/nginx-deployment image updated
PS D:\Desktop\stream>
```