

Lesson 02 Demo 04

Creating and Configuring the Deployment

Objective: To demonstrate the process of creating a Kubernetes deployment and accessing the associated pod

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

Steps to be followed:

1. Create the deployment
2. Access the pod

Step 1: Create the deployment

1.1 On the master node, execute the following command to create the deployment:

kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift

```
labsuser@master:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
master.example.com                  Ready     control-plane  175m  v1.28.2
worker-node-1.example.com           Ready     <none>      174m  v1.28.2
worker-node-2.example.com           Ready     <none>      171m  v1.28.2
labsuser@master:~$ kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift
deployment.apps/myapp1 created
labsuser@master:~$
```

- 1.2 Confirm the deployment's creation and inspect the state of the pods with the following commands:

kubectl get deployment

kubectl get pods

```
labsuser@master:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
master.example.com                  Ready    control-plane   175m   v1.28.2
worker-node-1.example.com           Ready    <none>         174m   v1.28.2
worker-node-2.example.com           Ready    <none>         171m   v1.28.2
labsuser@master:~$ kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift
deployment.apps/myapp1 created
labsuser@master:~$ kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
myapp1    1/1     1            1           53s
labsuser@master:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
myapp1-57bb57dd79-q6htr             1/1     Running    0           62s
labsuser@master:~$
```

You can see that both the deployment and the OpenShift pod have been successfully created and are operational.

Step 2: Access the pod

- 2.1 Expose the deployment to generate a service with the following command:

kubectl expose deployment myapp1 --port=8080

```
labsuser@master:~$ kubectl expose deployment myapp1 --port=8080
service/myapp1 exposed
labsuser@master:~$
```

Note: This step utilizes port **8080** to expose the deployment. Ensure other services do not occupy this port.

2.2 Run the following command to enumerate the services and pinpoint the ClusterIP:

kubectl get svc

```
labsuser@master:~$ kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    3h9m
myapp1        ClusterIP     10.101.186.24 <none>         8080/TCP    3m57s
labsuser@master:~$
```

Note: Save the ClusterIP for upcoming steps

2.3 Access the pod using the **curl** command and the previously saved ClusterIP:

curl <ClusterIP>:8080

```
labsuser@master:~$ kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    3h9m
myapp1        ClusterIP     10.101.186.24 <none>         8080/TCP    3m57s
labsuser@master:~$ curl 10.101.186.24:8080
Hello OpenShift!
labsuser@master:~$
```

You can see a greeting message from the OpenShift pod, confirming your access.

By following these steps, you have successfully created the Kubernetes deployment and accessed its associated pod.