**Lesson 2 Demo 5**

**Understanding Basic Commands of Kubernetes**

**Objective:** To execute basic commands used in Kubernetes

**Tools required:** kubeadm, kubectl

**Prerequisites: kubeadm** and **kubectl** should be installed

**Note 1:** This demo is based on Kubernetes version 1.23

Steps to be followed:

1. Creating the Deployment
2. Creating the namespaces
3. Scaling and deleting the Deployment

**Step 1: Creating the Deployment**

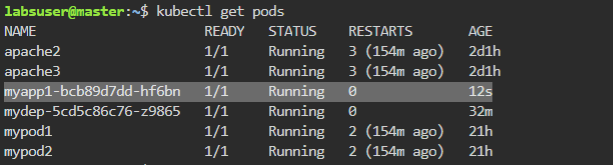
1. Run the following command to create the Deployment:

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



1. Verify the Deployment and Pods’ states by using the following command:

**kubectl get pods**



1. To describe Pod content, use the following command:

**kubectl describe pod myapp1-bcb89d7dd-q8jz5**

Text

Description automatically generated

Text

Description automatically generated

1. To verify the current running Deployment and describe content, use the following command:

**kubectl get deployment**

A screenshot of a computer

Description automatically generated

1. Replace the **deploymentName** with the name that you created:

**kubectl describe deployment <deploymentName>**Text

Description automatically generated

1. Create a YAML file by using the following command:

**kubectl create deployment myhttpd --image=docker.io/httpd --dry-run=client -o yaml > mydep.yaml**

1. Create the Deployment file by using the following command:

**vi mydep.yaml**

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

1. Expose the Deployment to create a service:

**kubectl expose deployment myapp1 --port=8080**

A screenshot of a computer

Description automatically generated

1. Verify the created services:

**kubectl get svc**

Graphical user interface, text, application

Description automatically generated

1. Describe the service by using the following command:

**kubectl describe svc myapp1**

Text

Description automatically generated

**Step 2: Creating the namespaces**

1. To create specific namespaces, use the following command:

**kubectl create namespace mynamespace**

Text

Description automatically generated

1. Verify the namespace:

**kubectl get namespace**

A screenshot of a computer

Description automatically generated

**Step 3: Scaling and deleting the Deployment**

1. To create a Deployment in a particular namespace, use the following command:

**kubectl create deployment mydep --image=docker.io/httpd -n mynamespace**

**kubectl get deployment -n namespace**

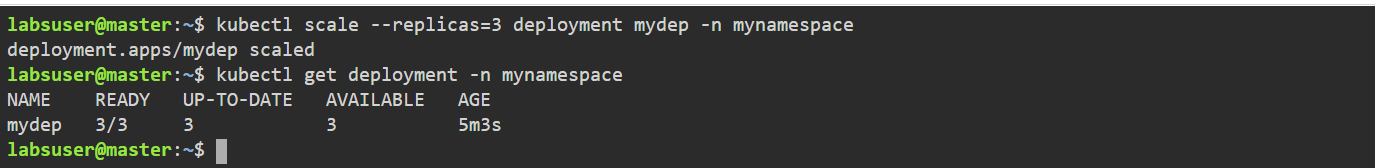
**kubectl get pods -n namespace**

A screenshot of a computer

Description automatically generated

1. Scale the Deployment by using the following command:

**kubectl scale --replicas=3 deployment mydep -n mynamespace**

**kubectl get deployment -n mynamespace**

1. Run the given command to get the endpoints:

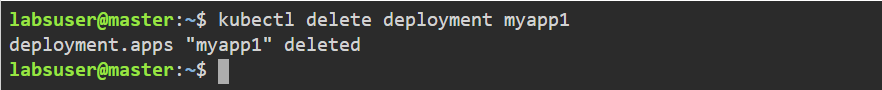
**kubectl get endpoints**

**kubectl describe endpoints**

Text

Description automatically generated

1. Delete the created Pods, Deployment, service, and namespace by using the following command:

**kubectl delete deployment myapp1**

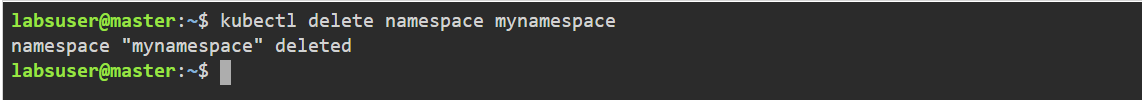
**kubectl delete svc mydep**

A screenshot of a computer

Description automatically generated with medium confidence

1. Delete the namespace:

**kubectl delete namespace mynamespace**



1. Verify the events by using following commands:

**kubectl get events**

Graphical user interface

Description automatically generated

1. Verify the Nodes’ state and describe Nodes’ configuration by using the following command:

**kubectl get nodes**

A screenshot of a computer

Description automatically generated

**kubectl describe node worker-node1.example.com**Text

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated