

# DevOps

## Kubernetes CLI Tool Assignment [CLO -01]

### Objective:

This assignment is designed to help you learn basic Kubernetes commands using the Kubernetes CLI (`kubectl`). By the end of this task, you will have hands-on experience in managing Kubernetes clusters and resources.

### Scenario:

Imagine you are a Kubernetes administrator tasked with setting up and managing a web application in a Kubernetes cluster. Using the `kubectl` CLI tool, perform the tasks outlined below and answer the related questions.

### Instructions:

- Complete all tasks step-by-step.
- Use `kubectl` commands as instructed.
- Document your answers and include screenshots wherever required.

### Part 1: Exploring the Cluster

1. **Verify Cluster Connectivity:** Run the following command to ensure your Kubernetes cluster is operational:

```
kubectl cluster-info
```

**Question:** What information does `kubectl cluster-info` provide? What services are listed?

2. **List Cluster Nodes:** Display the nodes in your cluster:

```
kubectl get nodes
```

### Questions:

- How many nodes are present?
- What is the status of each node?

## Part 2: Managing Pods

1. **List All Pods:** View all Pods running in the default namespace:

```
kubectl get pods
```

**Question:** What information is displayed (e.g., columns like NAME, READY, STATUS)?

2. **Inspect Pod Details:** Pick any Pod from the list and describe its configuration:

```
kubectl describe pod <pod-name>
```

**Question:** What details are provided in the `describe` output? Include details about events, labels, and container configurations.

3. **Delete a Pod:** Delete the Pod you just described:

```
kubectl delete pod <pod-name>
```

**Question:** If the Pod is part of a Deployment, what happens after deletion?

## Part 3: Managing Deployments

1. **Create a Deployment:** Create a Deployment for an `nginx` web server:

```
kubectl create deployment nginx-deployment --image=nginx
```

**Questions:**

- o How do you verify the Deployment was successfully created?
- o By default, how many replicas are created?

2. **Scale the Deployment:** Increase the number of replicas for the Deployment to 3:

```
kubectl scale deployment nginx-deployment --replicas=3
```

**Questions:**

- o How many Pods are now running?
- o How can you confirm the scaling was successful?

3. **Update the Deployment:** Update the Deployment to use the image `nginx:1.23`:

```
kubectl set image deployment/nginx-deployment nginx=nginx:1.23
```

### **Questions:**

- What command do you use to monitor the progress of this update?
- What happens to the Pods during the update process?

## **Part 4: Working with Services**

1. **Expose the Deployment:** Make the `nginx-deployment` accessible by exposing it as a Service:

```
kubectl expose deployment nginx-deployment --type=NodePort --port=80
```

### **Questions:**

- What is the purpose of exposing a Deployment as a Service?
- How do you find the external IP and port to access the service?

2. **Inspect the Service:** View the details of the created Service:

```
kubectl get services
```

### **Questions:**

- What is the `ClusterIP` of the Service?
- What is the difference between `ClusterIP` and `NodePort`?

## **Part 5: Clean-Up**

1. **Delete Resources:** Remove the Service and Deployment created during the assignment:

```
kubectl delete service nginx-deployment  
kubectl delete deployment nginx-deployment
```

### **Questions:**

- Why is it important to clean up unused resources?
- How can you confirm that all resources have been deleted?

## **Submission Guidelines:**

1. Document answers to all questions in a separate file.
2. Provide screenshots for the following:
  - Output of `kubectl get nodes`
  - Output of `kubectl describe pod`

- Output of `kubectl get services` showing the exposed Service
- 3. Include your command history to demonstrate task completion. Use the `history` command to capture it.

**Deadline:** December – 21 - 2024

**Good luck** Remember, Kubernetes is all about practice. Don't hesitate to experiment and explore additional commands