Kubernetes Pods Hands-On Lab — Single & Multi-Container

Author : Sandeep Kumar Sharma

Objective: By the end of this lab, participants will learn:

1. How to create a Pod using commands.
2. How to describe a Pod.
3. How to delete a Pod.
4. How to get inside (exec into) a Pod.
5. How to create a multi-container Pod using YAML.

Part 1: Working with Single Container Pod (Using Commands)

Step 1 — Create a Pod

Use the following command to create an Nginx Pod directly from the command line:

```
kubectl run mypod --image=nginx --restart=Never
```

Explanation:

- `kubectl run` creates a Pod.
- `--image=nginx` specifies the container image.
- `--restart=Never` ensures it creates a Pod (not a Deployment).

Step 2 — List Pods

To check the running Pods:

```
kubectl get pods
```

You'll see output like:

```
NAME     READY   STATUS    RESTARTS   AGE
mypod    1/1     Running   0          20s
```

Step 3 — Describe the Pod

Get detailed information about the Pod:

```
kubectl describe pod mypod
```

This command gives details such as IP, Node, Events, and container information.

Step 4 — Go Inside the Pod (Exec Command)

To access the Nginx Pod shell:

```
kubectl exec -it mypod -- /bin/bash
```

If `/bin/bash` is not available, use:

```
kubectl exec -it mypod -- /bin/sh
```

Once inside, you can check running processes or explore:

```
ls /usr/share/nginx/html
```

Type `exit` to come out of the container shell.

Step 5 — Delete the Pod

To remove the Pod:

```
kubectl delete pod mypod
```

You can verify:

```
kubectl get pods
```

It should now be gone.

Part 2: Creating a Pod Using YAML File

Step 1 — Create YAML Definition

Create a file named `nginx-pod.yaml`:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:latest
    ports:
    - containerPort: 80
```

Step 2 — Apply the YAML File

```
kubectl apply -f nginx-pod.yaml
```

Step 3 — Verify Pod Creation

```
kubectl get pods
kubectl describe pod nginx-pod
```

Step 4 — Access the Pod

You can use exec command again:

```
kubectl exec -it nginx-pod -- /bin/sh
```

Inside the container, check the Nginx default index file:

```
cat /usr/share/nginx/html/index.html
```

Exit using `exit`.

Step 5 — Delete the Pod

```
kubectl delete pod nginx-pod
```

Part 3: Multi-Container Pod (Using YAML)

Step 1 — Create YAML Definition

Now we'll create a Pod that runs **two containers** inside the same Pod — one Nginx web server and one busybox sidecar container that continuously prints logs.

Create a file named `multi-container-pod.yaml`:

```
apiVersion: v1
kind: Pod
metadata:
  name: multi-container-pod
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
  - name: sidecar-container
    image: busybox
    command: ['sh', '-c', 'while true; do echo Sidecar container is running; sleep 5; done']
```

Step 2 — Deploy the Pod

```
kubectl apply -f multi-container-pod.yaml
```

Step 3 — Verify Pod Creation

```
kubectl get pods
kubectl describe pod multi-container-pod
```

Check that there are two containers inside the same Pod.

Step 4 — Check Logs of Each Container

```
kubectl logs multi-container-pod -c nginx-container
kubectl logs multi-container-pod -c sidecar-container
```

Step 5 — Exec into Each Container

To go inside Nginx container:

```
kubectl exec -it multi-container-pod -c nginx-container -- /bin/sh
```

To go inside BusyBox sidecar container:

```
kubectl exec -it multi-container-pod -c sidecar-container -- /bin/sh
```

Step 6 — Delete the Pod

```
kubectl delete pod multi-container-pod
```

Verification Summary

Task	Command	Description
Create Pod (CLI)	`kubectl run mypod --image=nginx`	Creates a simple Pod
Describe Pod	`kubectl describe pod <pod>`	Shows details
Go inside Pod	`kubectl exec -it <pod> -- /bin/sh`	Access Pod shell
Delete Pod	`kubectl delete pod <pod>`	Removes the Pod
Multi-container Pod	`kubectl apply -f multi-container-pod.yaml`	Creates multi-container Pod

✅**Outcome:** After completing this lab, participants can confidently create, describe, and delete Pods using both CLI and YAML methods. They also understand how to work with multi-container Pods where containers share resources inside the same Pod.

Next Steps: Move to Service labs (ClusterIP, NodePort, LoadBalancer) to expose your Pods externally.