

Lab Exercise 8- Create POD in Kubernetes

Objective:

- Understand the basic structure and syntax of a Kubernetes Pod definition file (YAML).
- Learn to create, inspect, and delete a Pod in a Kubernetes cluster.

Prerequisites

- Kubernetes Cluster: You need a running Kubernetes cluster. You can set up a local cluster using tools like Minikube or kind, or use a cloud-based Kubernetes service.
- kubectl: Install and configure kubectl to interact with your Kubernetes cluster.
- Basic Knowledge of YAML: Familiarity with YAML format will be helpful as Kubernetes resource definitions are written in YAML.

Step-by-Step Guide

Step 1: Create a YAML File for the Pod

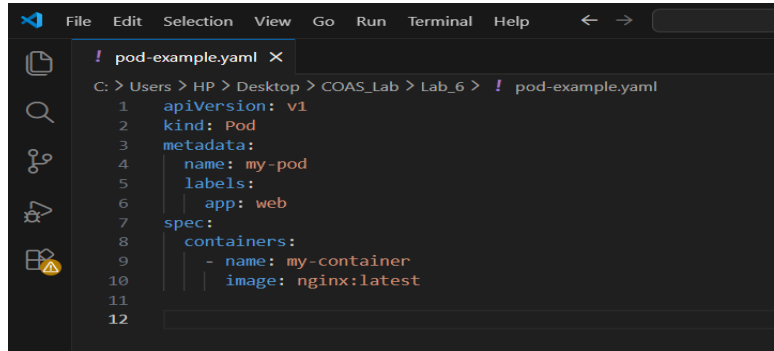
We'll create a Pod configuration file named **pod-example.yaml**

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
    app: web
spec:
  containers:
    - name: my-container
      image: nginx:latest
```

Explanation of the YAML File

- apiVersion: Specifies the version of the Kubernetes API to use. For Pods, it's typically v1.
- kind: The type of object being created. Here it's a Pod.
- metadata: Provides metadata about the object, including name and labels. The name must be unique within the namespace, and labels help in identifying and organizing Pods.
- spec: Contains the specifications of the Pod, including:

- containers: Lists all containers that will run inside the Pod. Each container needs:
 - name: A unique name within the Pod.
 - image: The Docker image to use for the container.
 - ports: The ports that this container exposes.
 - env: Environment variables passed to the container.



```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: my-pod
5   labels:
6     app: web
7 spec:
8   containers:
9     - name: my-container
10      image: nginx:latest
```

```
PS C:\Users\HP> cd C:\Users\HP\Desktop\COAS_Lab\Lab_6
PS C:\Users\HP\Desktop\COAS_Lab\Lab_6> wsl
docker-desktop: /tmp/docker-desktop-root/run/desktop/mnt/host/c/Users/HP/Desktop/COAS_Lab/Lab_6# ls
Lab Exercise 10.docx      Lab Exercise 7.docx
Lab Exercise 11.docx      Lab Exercise 8.docx
Lab Exercise 12.docx      Lab Exercise 9.docx
Lab Exercise 7- Install Minikube on Linux (Ubuntu -MacOS-Windows).docx  pod-example.yaml
Lab Exercise 7- Install Minikube on Linux (Ubuntu -MacOS-Windows).pdf
docker-desktop: /tmp/docker-desktop-root/run/desktop/mnt/host/c/Users/HP/Desktop/COAS_Lab/Lab_6# |
```

Step 2: Apply the YAML File to Create the Pod

Use the kubectl apply command to create the Pod based on the YAML configuration file.

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

This command tells Kubernetes to create a Pod as specified in the pod-example.yaml file.

```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_6> wsl -d Ubuntu-22.04 -u shivang
shivang@Shivang: /mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ shivang@Shivang:~$
shivang@Shivang:~$: command not found
shivang@Shivang: /mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped
```

```
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ minikube start --driver=docker
🐳 minikube v1.38.0 on Ubuntu 22.04 (kvm/amd64)
🔥 Using the docker driver based on existing profile
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.49 ...
🔄 Restarting existing docker container for "minikube" ...
🚨 StartHost failed, but will try again: driver start: start: docker start minikube: exit status 1
stdout:

stderr:
Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: unable to apply cgroup configuration: unable to start unit "docker-a8a10b6c11e9975e383ea94e53e8342187ad8c183c5006ed26beb0577de1ea55.scope" (properties [{Name:Description Value:"libcontainer container a8a10b6c11e9975e383ea94e53e8342187ad8c183c5006ed26beb0577de1ea55"} {Name:Slice Value:"system.slice"} {Name:Delegate Value:true} {Name:PIPs Value:@au [974]} {Name:MemoryAccounting Value:true} {Name:CPUCAccounting Value:true} {Name:IOAccounting Value:true} {Name:TasksAccounting Value:true} {Name:DefaultDependencies Value:false}]): error creating systemd unit 'docker-a8a10b6c11e9975e383ea94e53e8342187ad8c183c5006ed26beb0577de1ea55.scope': got 'failed': unknown
Error: failed to start containers: minikube

🔄 Restarting existing docker container for "minikube" ...
📦 Preparing Kubernetes v1.35.0 on Docker 29.2.0 ...
🔍 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🔥 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ |

shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl apply -f pod-example.yaml
pod/my-pod created
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ |
```

Step 3: Verify the Pod Creation

To check the status of the Pod and ensure it's running, use:

```
kubectl get pods
```

This command lists all the Pods in the current namespace, showing their status, restart count, and other details.

You can get detailed information about the Pod using:

```
kubectl describe pod my-pod
```

This command provides detailed information about the Pod, including its events, container specifications, and resource usage.

```

shivang@Shivang: /mnt/c/Usi x + v
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-pod    1/1     Running   0           80s
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl describe pod my-pod
Name:      my-pod
Namespace: default
Priority:   0
Service Account: default
Node:      minikube/192.168.49.2
Start Time: Mon, 09 Feb 2026 23:32:47 +0530
Labels:    app=web
Annotations: <none>
Status:    Running
IP:        10.244.0.5
IPs:
  IP: 10.244.0.5
Containers:
  my-container:
    Container ID:   docker://03c12e26ca37fca0e6c958798a739e8caaa4f9767751fe835b0f9a188cc579ed
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Mon, 09 Feb 2026 23:33:13 +0530
    Ready:          True
    Restart Count:   0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-9lvw8 (ro)
Conditions:
  Type              Status
  PodReadyToStartContainers  True
  Initialized         True
  Ready               True
  ContainersReady     True
  PodScheduled        True
Volumes:
  kube-api-access-9lvw8:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
Volumes:
  kube-api-access-9lvw8:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    Optional:           false
    DownwardAPI:        true
QoS Class:             BestEffort
Node-Selectors:        <none>
Tolerations:           node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                      node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age   From          Message
  ----    -
  Normal  Scheduled   92s   default-scheduler  Successfully assigned default/my-pod to minikube
  Normal  Pulling     92s   kubelet        spec.containers{my-container}: Pulling image "nginx:latest"
  Normal  Pulled      67s   kubelet        spec.containers{my-container}: Successfully pulled image "nginx:latest" in 25.185s (25.185s including waiting)
  Image size: 169850673 bytes.
  Normal  Created    67s   kubelet        spec.containers{my-container}: Container created
  Normal  Started    67s   kubelet        spec.containers{my-container}: Container started
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$

```

Step 4: Interact with the Pod

You can interact with the running Pod in various ways, such as accessing the logs or executing commands inside the container.

View Logs: To view the logs of the container in the Pod:

```
kubectl logs my-pod
```

Execute a Command: To run a command inside the container:

```
kubectl exec -it my-pod -- /bin/bash
```

The `-it` flag opens an interactive terminal session inside the container, allowing you to run commands.

```
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl logs my-pod
/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
2026/02/09 18:03:13 [notice] 1#1: using the "epoll" event method
2026/02/09 18:03:13 [notice] 1#1: nginx/1.29.5
2026/02/09 18:03:13 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/02/09 18:03:13 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2026/02/09 18:03:13 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/02/09 18:03:13 [notice] 1#1: start worker processes
2026/02/09 18:03:13 [notice] 1#1: start worker process 29
2026/02/09 18:03:13 [notice] 1#1: start worker process 30
2026/02/09 18:03:13 [notice] 1#1: start worker process 31
2026/02/09 18:03:13 [notice] 1#1: start worker process 32
2026/02/09 18:03:13 [notice] 1#1: start worker process 33
2026/02/09 18:03:13 [notice] 1#1: start worker process 34
2026/02/09 18:03:13 [notice] 1#1: start worker process 35
2026/02/09 18:03:13 [notice] 1#1: start worker process 36
2026/02/09 18:03:13 [notice] 1#1: start worker process 37
2026/02/09 18:03:13 [notice] 1#1: start worker process 38
2026/02/09 18:03:13 [notice] 1#1: start worker process 39
2026/02/09 18:03:13 [notice] 1#1: start worker process 40
2026/02/09 18:03:13 [notice] 1#1: start worker process 41
2026/02/09 18:03:13 [notice] 1#1: start worker process 42
2026/02/09 18:03:13 [notice] 1#1: start worker process 43
2026/02/09 18:03:13 [notice] 1#1: start worker process 44
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl exec -it my-pod -- /bin/bash
root@my-pod:/# echo "Hello I am Shivang"
Hello I am Shivang
root@my-pod:/#
root@my-pod:/# exit
exit
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ |
```

Step 5: Delete the Pod

To clean up and remove the Pod when you're done, use the following command:

```
kubectl delete pod my-pod
```

This command deletes the specified Pod from the cluster.

```
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl delete pod my-pod
pod "my-pod" deleted from default namespace
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ kubectl get pods
No resources found in default namespace.
shivang@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_6$ |
```