

Lesson 05 Demo 04

Creating and Configuring Pod Priority and Preemption

Objective: To demonstrate the creation and configuration of priority classes and assigning them to pods in a Kubernetes environment

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 and describe the priority class object
2. Create and describe the pod priority file

Step 1: Create and describe the priority class object

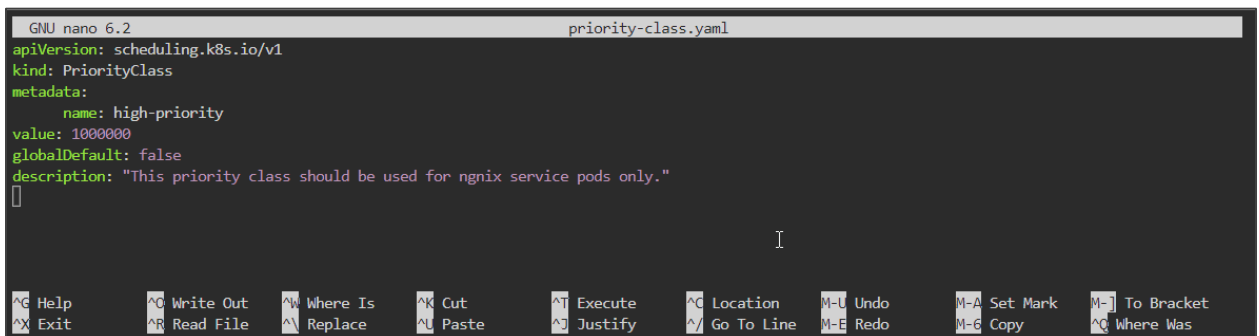
1.1 Create the YAML file by using the following command:

```
nano priority-class.yaml
```

```
labsuser@master:~$ nano priority-class.yaml
```

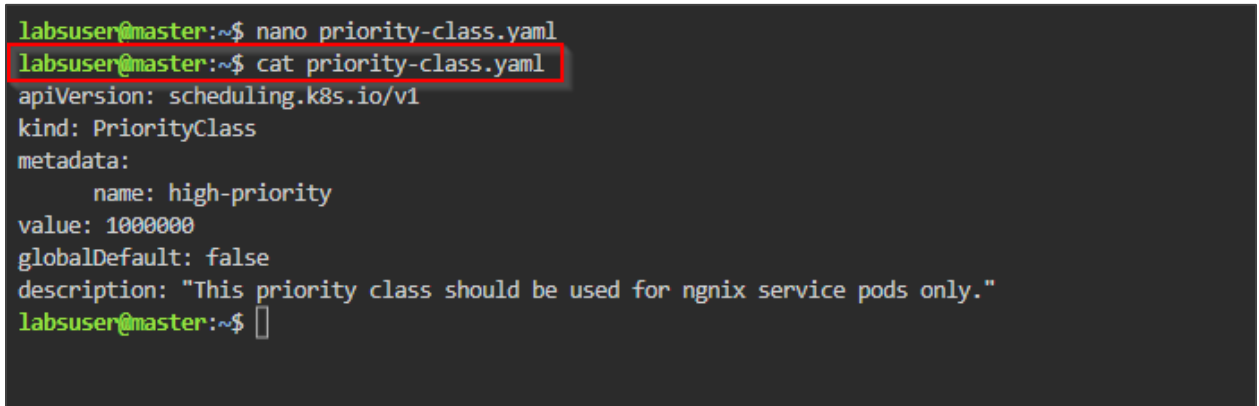
1.2 Add the following code to the **priority-class.yaml** file:

```
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "This priority class should be used for nginx service pods only."
```



```
GNU nano 6.2 priority-class.yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "This priority class should be used for nginx service pods only."
[]
```

1.3 Use the **cat** command to validate the content of the **priority-class.yaml** file



```
labsuser@master:~$ nano priority-class.yaml
labsuser@master:~$ cat priority-class.yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "This priority class should be used for nginx service pods only."
labsuser@master:~$ []
```

1.4 Create the priority-class resource by using the following command:

kubectl apply -f priority-class.yaml

```
labsuser@master:~$ cat priority-class.yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
value: 1000000
globalDefault: false
description: "This priority class should be used for nginx service pods only."
labsuser@master:~$ kubectl apply -f priority-class.yaml
priorityclass.scheduling.k8s.io/high-priority created
labsuser@master:~$
```

1.5 List the available priority classes by using the following command:

kubectl get priorityclasses

```
labsuser@master:~$ kubectl apply -f priority-class.yaml
priorityclass.scheduling.k8s.io/high-priority created
labsuser@master:~$ kubectl get priorityclasses
```

NAME	VALUE	GLOBAL-DEFAULT	AGE
high-priority	1000000	false	3m46s
system-cluster-critical	2000000000	false	3d2h
system-node-critical	2000001000	false	3d2h

```
labsuser@master:~$
```

1.6 Describe the created priority classes by using the following command:

kubectl describe priorityclasses high-priority

```
system-node-critical    2000001000    false    3d2h
labsuser@master:~$ kubectl describe priorityclasses high-priority
```

```
Name:                high-priority
Value:               1000000
GlobalDefault:      false
PreemptionPolicy:    PreemptLowerPriority
Description:         This priority class should be used for nginx service pods only.
Annotations:         kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"scheduling.k8s.io/v1","description":"This priority class should be used for nginx service pods only.\"", "globalDefault":false,"kind":"PriorityClass","metadata":{"annotations":{},"name":"high-priority"},"value":1000000}
Events:              <none>
labsuser@master:~$
```

Step 2: Create and describe the pod priority file

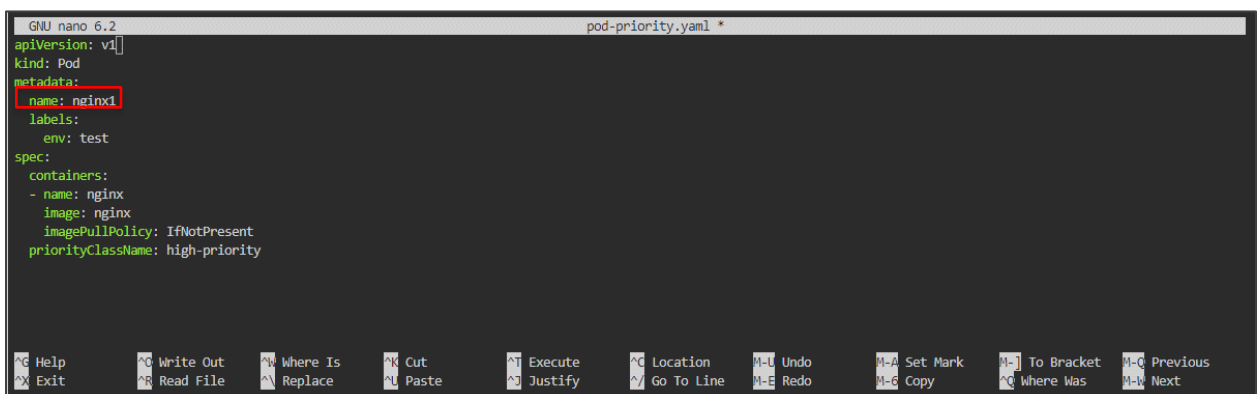
2.1 Create the YAML file by using the following command:

nano pod-priority.yaml

```
labsuser@master:~$ kubectl describe priorityclasses high-priority
Name:          high-priority
Value:         1000000
GlobalDefault: false
PreemptionPolicy: PreemptLowerPriority
Description:    This priority class should be used for nginx service pods only.
Annotations:    kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"scheduling.k8s.io/v1","description":"This priority class should be used for nginx service pods only.", "globalDefault":false,"kind":"PriorityClass","metadata":{"annotations":{},"name":"high-priority"},"value":1000000}
Events:         <none>
labsuser@master:~$ nano pod-priority.yaml
```

2.2 Add the following code to the **pod-priority.yaml** file:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  priorityClassName: high-priority
```



```
GNU nano 6.2 pod-priority.yaml *
apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  priorityClassName: high-priority

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^J Execute   ^C Location  ^U Undo      ^I-A Set Mark ^I= To Bracket ^I-Q Previous
^X Exit      ^R Read File ^\ Replace   ^N Paste     ^_ Justify   ^_ Go To Line ^-E Redo     ^I-E Copy     ^I_Q Where Was ^I-W Next
```

Note: If a pod with the name **nginx** has already been created, you may need to choose a different name for the pod, as indicated in the screenshot above.

2.3 Use the **cat** command to validate the content of the **pod-priority.yaml** file

```
labsuser@master:~$ nano pod-priority.yaml
labsuser@master:~$ cat pod-priority.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
    priorityClassName: high-priority
labsuser@master:~$
```

2.4 Create the pod-priority class resource by using the following command:

kubectl apply -f pod-priority.yaml

```
labsuser@master:~$ nano pod-priority.yaml
labsuser@master:~$ cat pod-priority.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
    priorityClassName: high-priority
labsuser@master:~$ kubectl apply -f pod-priority.yaml
pod/nginx1 created
labsuser@master:~$
```

2.5 List the available pods by using the following command:

kubectl get pods

```
labsuser@master:~$ kubectl apply -f pod-priority.yaml
pod/nginx1 created
labsuser@master:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
frontend-7hnr1	1/1	Running	3 (3h1m ago)	2d
frontend-cf7tz	1/1	Running	3 (19h ago)	2d
mydep-548c7db5df-dsvk8	0/1	CreateContainerError	0	2d
mydep-6f74bcdf49-dh2vc	0/1	CreateContainerError	0	2d
nginx	1/1	Running	0	18m
nginx1	1/1	Running	0	2m43s
security-context-1	1/1	Running	9 (47s ago)	24h

```
labsuser@master:~$
```

2.6 Describe the pod to verify the priority class by using the following command:

kubectl describe pod nginx1

```
labsuser@master:~$ kubectl describe pod nginx1
```

Name: nginx1
Namespace: default
Priority: 1000000
Priority Class Name: high-priority
Service Account: default
Node: worker-node-2.example.com/172.31.25.241
Start Time: Thu, 12 Oct 2023 10:31:32 +0000
Labels: env=test
Annotations: cni.projectcalico.org/containerID: 49c2d68f3e8367465cd58bcabe84984adac226ba873776226c1d2f70e360c9ca
cni.projectcalico.org/podIP: 192.168.232.200/32
cni.projectcalico.org/podIPs: 192.168.232.200/32

Status: Running
IP: 192.168.232.200
IPs: IP: 192.168.232.200
Containers:
 nginx:
 Container ID: containerd://550a2854e39a02b44e13c7cc065ebc295b672fa84bf983ee5f4cd7e5d431f0ae
 Image: nginx

```
Volumes:
  kube-api-access-wrr5z:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    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   8m1s   default-scheduler   Successfully assigned default/nginx1 to worker-node-2.example.com
Normal    Pulling     8m1s   kubelet         Pulling image "nginx"
Normal    Pulled      7m57s   kubelet         Successfully pulled image "nginx" in 3.988s (3.988s including waiting)
Normal    Created     7m57s   kubelet         Created container nginx
Normal    Started     7m57s   kubelet         Started container nginx
labsuser@master:~$
```

By following these steps, you have successfully configured priority classes and associated them with the pods in a Kubernetes environment, ensuring efficient workload management and resource allocation.