Lesson 08 Demo 02

Creating a Pod in Kubernetes

Objective: To demonstrate the creation of multi-container and single-container pods in Kubernetes, along with resolving pending state issues by removing node taints

Tools required: Docker, Kubernetes

Pre-requisites: Lesson 08 Demo 01

Steps to be followed:

1. Create multi-container pods
2. Create a single container pod

Step 1: Create multi-container pods

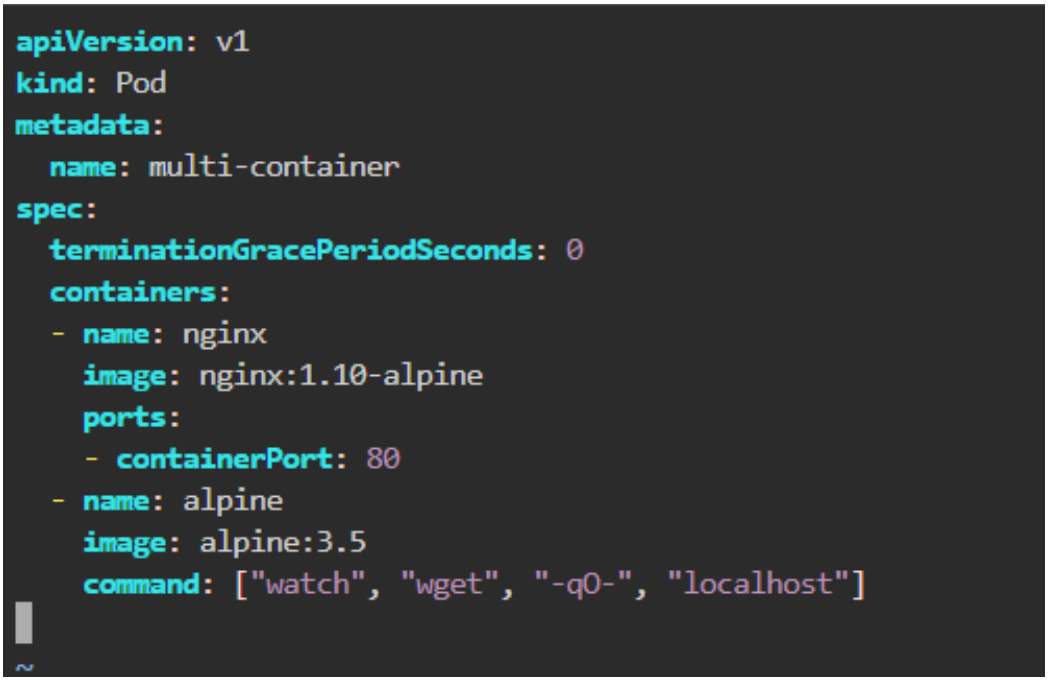
1.1 Navigate to the master node and create a new file named **sample.yaml:**
sudo su
vi sample.yaml

```
labsuser@ip-172-31-41-222:~$ sudo su
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml

[1]+  Stopped                 vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# kubectl create -f sample.yaml
pod/multi-container created
```

1.2 Add the following code to the **sample.yaml** file:

```
apiVersion: v1
kind: Pod
metadata:
 name: multi-container
spec:
 terminationGracePeriodSeconds: 0
 containers:
 - name: nginx
  image: nginx:1.10-alpine
  ports:
  - containerPort: 80
 - name: alpine
  image: alpine:3.5
  command: ["watch", "wget", "-qO-", "localhost"]
```



1.3 Use the following command to create the multi-container pod:

kubectl create -f sample.yaml

```
labsuser@ip-172-31-41-222:~$ sudo su
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml

[1]+  Stopped                 vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# kubectl create -f sample.yaml
pod/multi-container created
```

Step 2: Create a single container pod

2.1 On the master node, create a single container pod with a Tomcat image using the following command, and check the pods:

kubectl run tomcat --image=tomcat:8.0

kubectl get pods

```
labsuser@ip-172-31-41-222:~$ sudo su
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml

[1]+  Stopped                 vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# vi sample.yaml
root@ip-172-31-41-222:/home/labsuser# kubectl create -f sample.yaml
pod/multi-container created
root@ip-172-31-41-222:/home/labsuser# kubectl run tomcat --image=tomcat:8.0
pod/tomcat created
root@ip-172-31-41-222:/home/labsuser# kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
multi-container   0/2     Pending   0          23s
tomcat            0/1     Pending   0          11s
```

2.2 Run the following command to check the pod in its pending state:
kubectl describe pods <pod_name>

```
root@ip-172-31-41-222:/home/labsuser# kubectl describe pods multi-container
Name:         multi-container
Namespace:    default
Priority:     0
Node:         <none>
Labels:       <none>
Annotations:  <none>
Status:       Pending
IP:
IPs:          <none>
Containers:
  nginx:
    Image:        nginx:1.10-alpine
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-lq8p6 (ro)
  alpine:
    Image:      alpine:3.5
    Port:       <none>
    Host Port:  <none>
    Command:
```

2.3 To get the nodes status, use the following command:
kubectl get nodes

```
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  59s   default-scheduler  0/1 nodes are available: 1 node(s) had taint {node-role.ku
didn't tolerate.
  Warning  FailedScheduling  59s   default-scheduler  0/1 nodes are available: 1 node(s) had taint {node-role.ku
didn't tolerate.
root@ip-172-31-41-222:/home/labsuser# kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
multi-container   0/2     Pending   0          4m24s
tomcat            0/1     Pending   0          4m12s
root@ip-172-31-41-222:/home/labsuser# kubectl taint nodes multi-container  node-role.kubernetes.io/master-
Error from server (NotFound): nodes "multi-container" not found
root@ip-172-31-41-222:/home/labsuser# kubectl get nodes
NAME               STATUS   ROLES                  AGE   VERSION
ip-172-31-41-222   Ready    control-plane,master   87m   v1.20.5
```

2.4 To remove the taint from the node, run the following commands:
kubectl taint nodes <node name> node-role.kubernetes.io/master-

```
root@ip-172-31-41-222:/home/labsuser# kubectl get nodes
NAME               STATUS   ROLES                  AGE   VERSION
ip-172-31-41-222   Ready    control-plane,master   87m   v1.20.5
root@ip-172-31-41-222:/home/labsuser# kubectl taint nodes ip-172-31-41-222  node-role.kubernetes.io/master-
node/ip-172-31-41-222 untainted
```

Note: Copy the node name and use it in the command.

2.5 Now, check the pod status by using the below command:
sudo kubectl get pods

```
root@ip-172-31-41-222:/home/labsuser# kubectl taint nodes ip-172-31-41-222  node-role.kubernetes.io/master-
node/ip-172-31-41-222 untainted
root@ip-172-31-41-222:/home/labsuser# sudo kubectl get pods
NAME              READY   STATUS              RESTARTS   AGE
multi-container   0/2     ContainerCreating   0          7m2s
tomcat            0/1     ContainerCreating   0          6m50s
```

You can see that the pods are in ready state.

By following these steps, you have successfully created multi-container and single container pods in Kubernetes, and resolved any pending state issues by removing the taint from the node.