Lesson 07 Demo 07

Troubleshooting an Application Pod in Kubernetes

Objective: To set up an application pod in Kubernetes, diagnose potential issues, and implement necessary troubleshooting steps to ensure successful application deployment

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

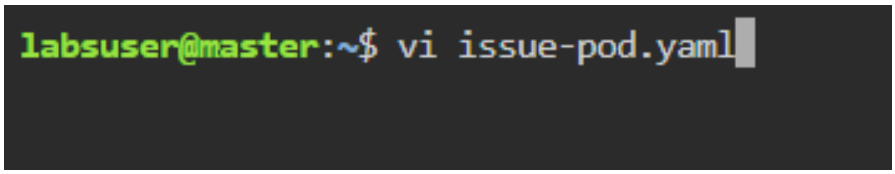
Steps to be followed:

1. Set up and diagnose the application pod

Step 1: Set up and diagnose the application pod

1.1 Run the following command to create an **issue-pod.yaml** file:

vi issue-pod.yaml



1.2 To create a deployment, add the following YAML code to the **issue-pod.yaml** file:

```
apiVersion: v1
kind: Pod
metadata:
  name: openshift
  labels:
    Podlabel: simplilearn
spec:
  containers:
  - name: mycontainer
    image: docker.io/openshift
    ports:
    - containerPort: 80
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: openshift
  labels:
    Podlabel: simplilearn
spec:
  containers:
  - name: mycontainer
    image: docker.io/openshift
    ports:
    - containerPort: 80
```

1.3 Deploy the **issue-pod.yaml** file using the following command:
kubectl create -f issue-pod.yaml

```
labsuser@master:~$ vi issue-pod.yaml
labsuser@master:~$ kubectl create -f issue-pod.yaml
pod/openshift created
labsuser@master:~$
```

1.4 To verify the pods, run the following command:
kubectl get pods

```
labsuser@master:~$ vi issue-pod.yaml
labsuser@master:~$ kubectl create -f issue-pod.yaml
pod/openshift created
labsuser@master:~$ kubectl get pods
NAME                          READY   STATUS             RESTARTS        AGE
frontend-6xkgb                1/1     Running            3 (3h32m ago)   28h
frontend-7q6qg                1/1     Running            3 (3h32m ago)   28h
frontend-bltgs                1/1     Running            3 (3h32m ago)   28h
mysql-7748c687bf-n9gdf        1/1     Running            1 (3h32m ago)   5h12m
nginx-7854ff8877-ktgkp        1/1     Running            0               166m
openshift                     0/1     ImagePullBackOff   0               113s
php-apache-5f9f45d488-d4lv7   1/1     Running            2 (3h32m ago)   27h
pod-env-var                   1/1     Running            3 (3h32m ago)   28h
pod-env12                     1/1     Running            3 (3h32m ago)   28h
testconfig                    0/1     Unknown            0               28h
wordpress-6ff4d555d5-tglfv    1/1     Running            1 (3h32m ago)   5h6m
labsuser@master:~$
```

1.5 To retrieve and display the events that have occurred within the Kubernetes cluster, use the following command:

kubectl get events

```
labsuser@master:~$ kubectl create -f issue-pod.yaml
pod/openshift created
labsuser@master:~$ kubectl get pods
NAME                        READY   STATUS             RESTARTS        AGE
frontend-6xkgb              1/1     Running            3 (3h32m ago)   28h
frontend-7q6qg              1/1     Running            3 (3h32m ago)   28h
frontend-bltgs              1/1     Running            3 (3h32m ago)   28h
mysql-7748c687bf-n9gdf      1/1     Running            1 (3h32m ago)   5h12m
nginx-7854ff8877-ktgkp      1/1     Running            0               166m
openshift                   0/1     ImagePullBackOff   0               113s
php-apache-5f9f45d488-d4lv7 1/1     Running            2 (3h32m ago)   27h
pod-env-var                 1/1     Running            3 (3h32m ago)   28h
pod-env12                   1/1     Running            3 (3h32m ago)   28h
testconfig                  0/1     Unknown            0               28h
wordpress-6ff4d555d5-tglfv  1/1     Running            1 (3h32m ago)   5h6m
labsuser@master:~$ kubectl get events
LAST SEEN   TYPE      REASON                   OBJECT                                MESSAGE
3m          Normal    Scheduled                pod/openshift                         Successfully assigned default/openshift to worker-node-2.example
.com
98s         Normal    Pulling                  pod/openshift                         Pulling image "docker.io/openshift"
97s         Warning   Failed                   pod/openshift                         Failed to pull image "docker.io/openshift": failed to pull and u
npack image "docker.io/library/openshift:latest": failed to resolve reference "docker.io/library/openshift:latest": pull access denied, repository d
oes not exist or may require authorization: server message: insufficient_scope: authorization failed
97s         Warning   Failed                   pod/openshift                         Error: ErrImagePull
70s         Normal    BackOff                  pod/openshift                         Back-off pulling image "docker.io/openshift"
84s         Warning   Failed                   pod/openshift                         Error: ImagePullBackOff
100s        Warning   FailedGetResourceMetric  horizontalpodautoscaler/wordpress     failed to get cpu utilization: missing request for cpu in contai
ner wordpress of Pod wordpress-6ff4d555d5-tglfv
labsuser@master:~$
```

1.6 To retrieve the details of the pod, use the following command:

kubectl describe pod openshift

```
labsuser@master:~$ kubectl describe pod openshift
Name:             openshift
Namespace:        default
Priority:         0
Service Account:  default
Node:             worker-node-2.example.com/172.31.26.113
Start Time:       Fri, 13 Oct 2023 15:58:26 +0000
Labels:           Podlabel=simplilearn
Annotations:      cni.projectcalico.org/containerID: e2d2f52bc19fe09ec31927eb685b39450882747bb492b67c18f0ea23a84185e9
                  cni.projectcalico.org/podIP: 192.168.232.218/32
                  cni.projectcalico.org/podIPs: 192.168.232.218/32
Status:           Pending
IP:               192.168.232.218
IPs:
  IP:  192.168.232.218
Containers:
  mycontainer:
    Container ID:
    Image:          docker.io/openshift
    Image ID:
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       ImagePullBackOff
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-qp5xm (ro)
```

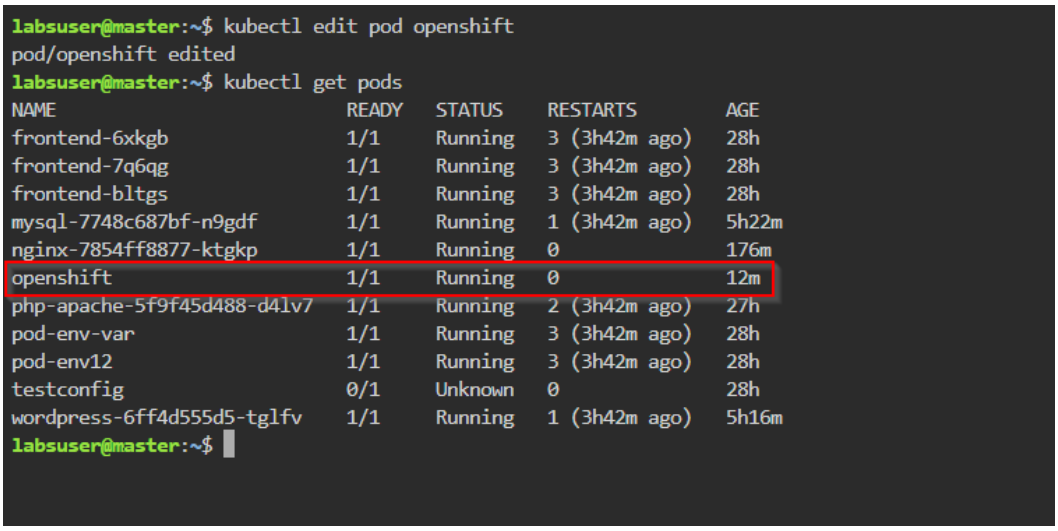
1.7 Change the service image for the pod from **docker.io/openshift** to **openshift/hello-openshift** using the following command:
kubectl edit pod openshift

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: Pod
metadata:
  annotations:
    cni.projectcalico.org/containerID: e2d2f52bc19fe09ec31927eb685b39450882747bb492b67c18f0ea23a84185e9
    cni.projectcalico.org/podIP: 192.168.232.218/32
    cni.projectcalico.org/podIPs: 192.168.232.218/32
  creationTimestamp: "2023-10-13T15:58:26Z"
  labels:
    Podlabel: simplilearn
  name: openshift
  namespace: default
  resourceVersion: "39792"
  uid: fa45a18b-db11-4bfb-bea8-f206fdee23ec
spec:
  containers:
  - image: docker.io/openshift
    imagePullPolicy: Always
    name: mycontainer
    ports:
    - containerPort: 80
      protocol: TCP
    resources: {}
    terminationMessagePath: /dev/termination-log
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    cni.projectcalico.org/containerID: e2d2f52bc19fe09ec31927eb685b39450882747bb492b67c18f0ea23a841
    cni.projectcalico.org/podIP: 192.168.232.218/32
    cni.projectcalico.org/podIPs: 192.168.232.218/32
  creationTimestamp: "2023-10-13T15:58:26Z"
  labels:
    Podlabel: simplilearn
  name: openshift
  namespace: default
  resourceVersion: "39792"
  uid: fa45a18b-db11-4bfb-bea8-f206fdee23ec
spec:
  containers:
  - image: openshift/hello-openshift
    imagePullPolicy: Always
    name: mycontainer
    ports:
    - containerPort: 80
      protocol: TCP
    resources: {}
    terminationMessagePath: /dev/termination-log
-- INSERT --
```

1.8 To confirm the changes in the pods, execute the following command:

kubectl get pods



As shown in the screenshot above, the status of the pod is now **Running**.

By following these steps, you have successfully set up a Kubernetes pod, gained an understanding of its operational state, and effectively troubleshot the observed issue.