Multi-Container Pod

Introduction:

A Pod can have multiple containers which will be sharing the same network and volumes.

Objectives:

1. Create a multi-container pod
2. Set the image of a container inside a pod
3. Access both the containers inside a pod
4. Delete the pod

1. Create a multi-container pod

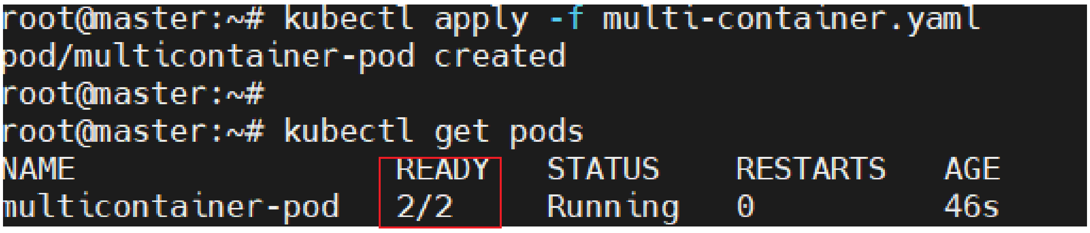
   We will be using the below mentioned YAML file to create the multi-container pod.

   ```
   # vi multi-container.yaml
   ```

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: multicontainer-pod
   spec:
     containers:
       - name: container1
         image: httpd:2.4
       - name: container2
         image: redis
   ```

   Apply the changes and pod will be created.

   ```
   Kubectl apply -f multi-container.yaml
   ```

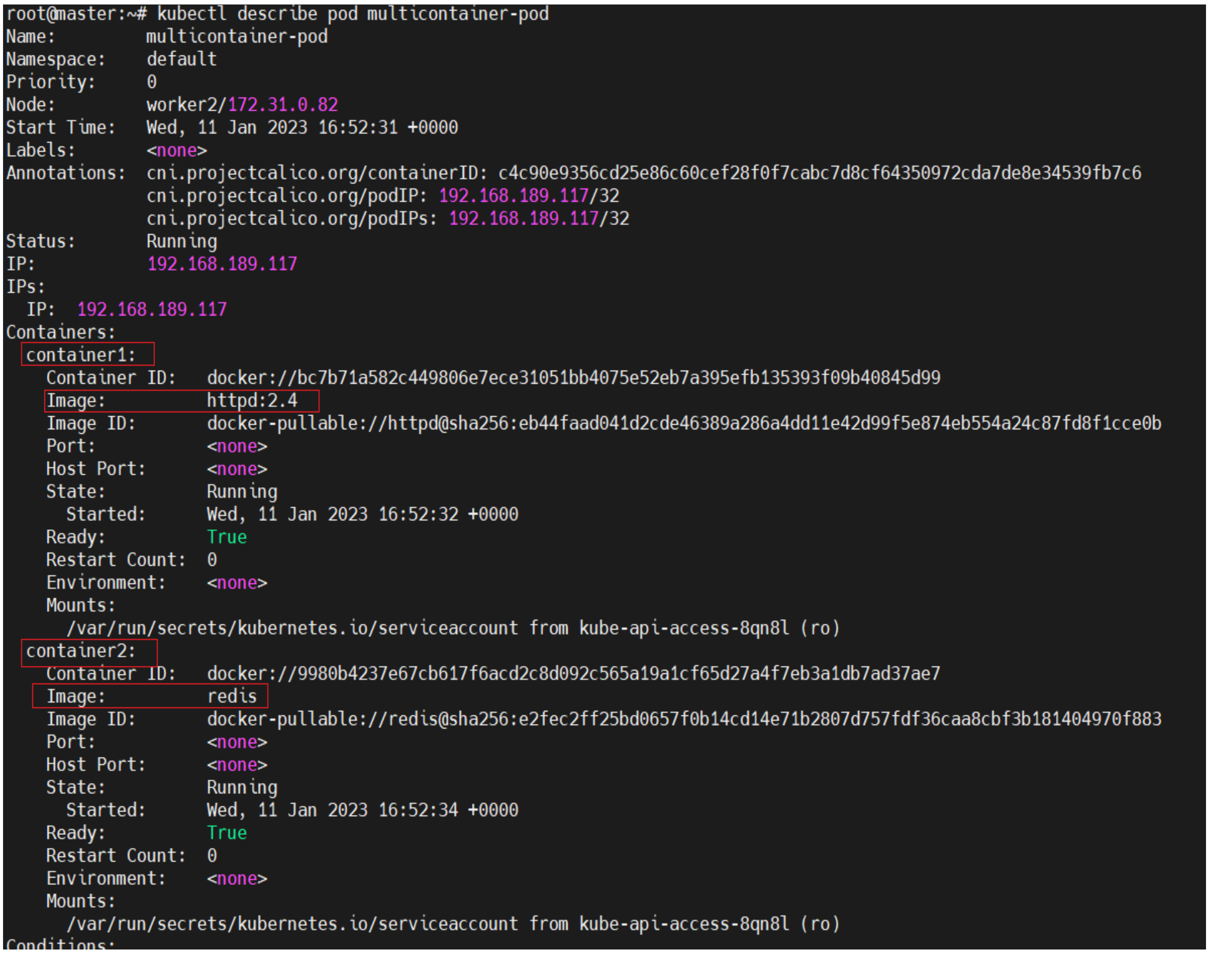


```
root@master:~# kubectl apply -f multi-container.yaml
pod/multicontainer-pod created
root@master:~#
root@master:~# kubectl get pods
NAME                 READY   STATUS    RESTARTS   AGE
multicontainer-pod   2/2     Running   0          46s
```

The above output shows that the pod has been created and 2 containers are ready inside this pod.

Further we can get more details using the description command.

```
kubectl describe pod multicontainer-pod
```

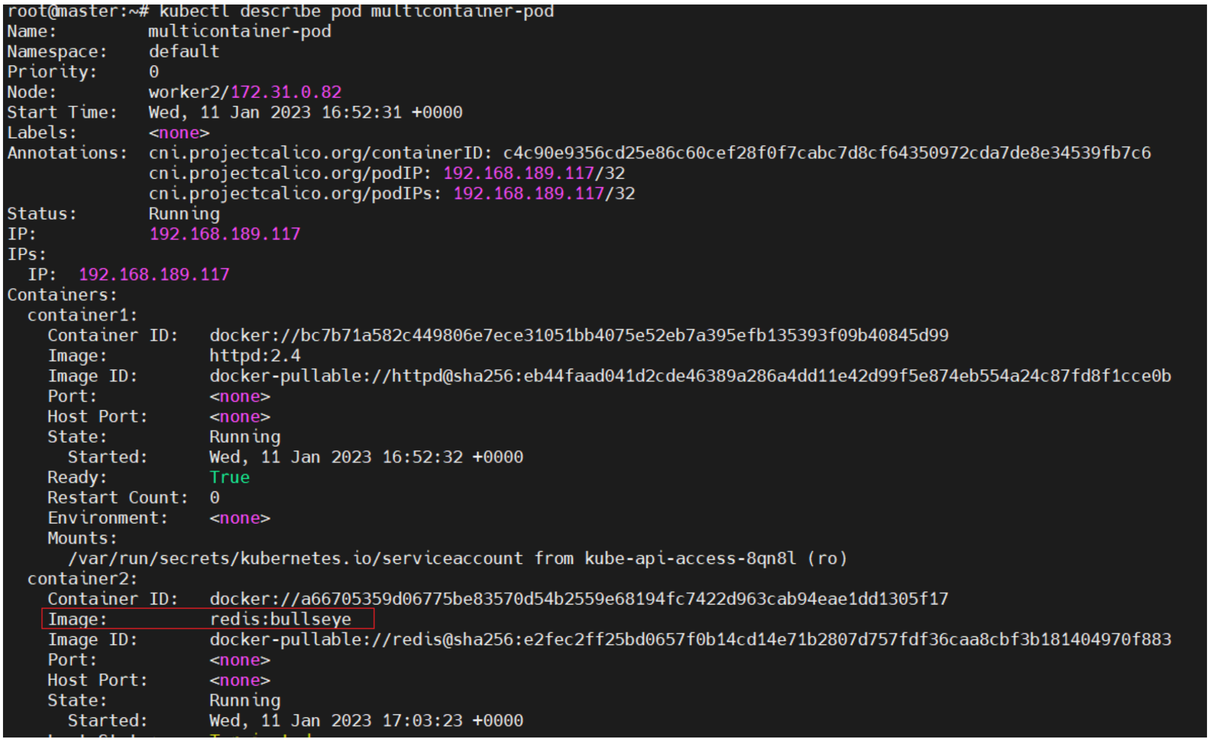


```
root@master:~# kubectl describe pod multicontainer-pod
Name:         multicontainer-pod
Namespace:    default
Priority:     0
Node:         worker2/172.31.0.82
Start Time:   Wed, 11 Jan 2023 16:52:31 +0000
Labels:       <none>
Annotations:  cni.projectcalico.org/containerID: c4c90e9356cd25e86c60cef28f0f7cabc7d8cf64350972cda7de8e34539fb7c6
              cni.projectcalico.org/podIP: 192.168.189.117/32
              cni.projectcalico.org/podIPs: 192.168.189.117/32
Status:       Running
IP:           192.168.189.117
IPs:
  IP:  192.168.189.117
Containers:
  container1:
    Container ID:   docker://bc7b71a582c449806e7ece31051bb4075e52eb7a395efb135393f09b40845d99
    Image:          httpd:2.4
    Image ID:       docker-pullable://httpd@sha256:eb44faad041d2cde46389a286a4dd11e42d99f5e874eb554a24c87fd8f1cce0b
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 11 Jan 2023 16:52:32 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8qn8l (ro)
  container2:
    Container ID:   docker://9980b4237e67cb617f6acd2c8d092c565a19a1cf65d27a4f7eb3a1db7ad37ae7
    Image:          redis
    Image ID:       docker-pullable://redis@sha256:e2fec2ff25bd0657f0b14cd14e71b2807d757fdf36caa8cbf3b181404970f883
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 11 Jan 2023 16:52:34 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8qn8l (ro)
Conditions:
```

2. <u>Set the image of a container inside a pod:</u>

Use the below command to set the image, we are changing the image of container 2 to **redis:bullseye**

```
kubectl set image pod multicontainer-pod container2=redis:bullseye
```



```
root@master:~# kubectl describe pod multicontainer-pod
Name:         multicontainer-pod
Namespace:    default
Priority:     0
Node:         worker2/172.31.0.82
Start Time:   Wed, 11 Jan 2023 16:52:31 +0000
Labels:       <none>
Annotations:  cni.projectcalico.org/containerID: c4c90e9356cd25e86c60cef28f0f7cabc7d8cf64350972cda7de8e34539fb7c6
              cni.projectcalico.org/podIP: 192.168.189.117/32
              cni.projectcalico.org/podIPs: 192.168.189.117/32
Status:       Running
IP:           192.168.189.117
IPs:
  IP:  192.168.189.117
Containers:
  container1:
    Container ID:   docker://bc7b71a582c449806e7ece31051bb4075e52eb7a395efb135393f09b40845d99
    Image:          httpd:2.4
    Image ID:       docker-pullable://httpd@sha256:eb44faad041d2cde46389a286a4dd11e42d99f5e874eb554a24c87fd8f1cce0b
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 11 Jan 2023 16:52:32 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8qn8l (ro)
  container2:
    Container ID:   docker://a66705359d06775be83570d54b2559e68194fc7422d963cab94eae1dd1305f17
    Image:          redis:bullseye
    Image ID:       docker-pullable://redis@sha256:e2fec2ff25bd0657f0b14cd14e71b2807d757fdf36caa8cbf3b181404970f883
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 11 Jan 2023 17:03:23 +0000
```

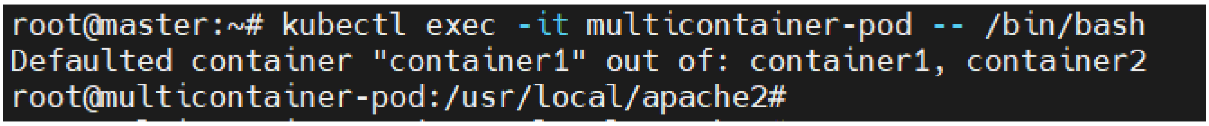
3. Access both the containers inside a pod:

We will be using the exec command to get inside the container. If do not mention the container then we would go to the first container automatically. Follow the below commands to enter inside a container.

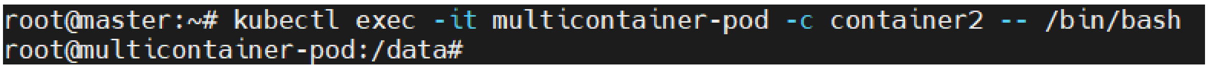
```
kubectl exec -it multicontainer-pod -- /bin/bash
kubectl exec -it multicontainer-pod -c container2 -- /bin/bash
```

The first command will let us enter inside the **container1** whereas the second command we have to mention the container name, in our case we have **container2** in this **multicontainer-pod**.

Refer the below output.



```
root@master:~# kubectl exec -it multicontainer-pod -- /bin/bash
Defaulted container "container1" out of: container1, container2
root@multicontainer-pod:/usr/local/apache2#
```



```
root@master:~# kubectl exec -it multicontainer-pod -c container2 -- /bin/bash
root@multicontainer-pod:/data#
```

4. Delete the Pod:

Use the below command to delete the pod

```
Kubectl delete pod multicontainer-pod
```