### **Lab Exercise 8- initContainer in Pod in Kubernetes**

In Kubernetes, an initContainer is a special type of container that runs and completes before the main application containers start. initContainers can contain utilities or setup scripts not present in the main application container image. They ensure that certain tasks are completed before the main application containers start.

Here's a step-by-step lab exercise to understand how initContainer works in Kubernetes.

**Step 1: Set Up Kubernetes Cluster**

* Ensure you have access to a Kubernetes cluster. You can use a local setup with Minikube, kind, or use a cloud-based Kubernetes service.

**Step 2: Create a Pod Manifest with an initContainer**

* Create a new file named **pod-init-container.yaml**.
* Define a Pod with an initContainer and a main application container. Here is an example manifest:

apiVersion: v1

kind: Pod

metadata:

name: init-container-demo

spec:

initContainers:

- name: init-myservice

image: busybox

command: ['sh', '-c', 'echo The init container is running and setting up; sleep 30']

containers:

- name: myapp-container

image: busybox

command: ['sh', '-c', 'echo The main application container is running; sleep 30’]

**This manifest defines a Pod with:**

* An initContainer named init-myservice that runs a command to print a message and sleep for 30 seconds.
* A main application container named myapp-container that prints a message and then sleeps for an hour.

**Step 3: Apply the Pod Manifest**

Run the following command to create the Pod in your Kubernetes cluster:

kubectl apply -f pod-init-container.yaml

**Step 4: Verify the Pod Creation**

Check the status of the Pod to ensure the initContainer runs and completes before the main application container starts.

kubectl get pods -w

You should see the Pod with a status indicating it is running or completed.

**Step 5: Inspect Pod Logs**

Fetch the logs for the initContainer and the main application container to verify their execution:

Get the logs for the initContainer:

kubectl logs init-container-demo -c init-myservice

You should see the message from the initContainer.

Get the logs for the main application container:

kubectl logs init-container-demo -c myapp-container

You should see the message from the main application container.

**Step 6: Clean Up**

After completing the exercise, clean up the resources created:

kubectl delete pod init-container-demo