Lesson 4 Demo 20: InitContainers Demonstration

This section will guide you to:

* Demonstrate InitContainers

This lab has one sub-section, namely:

1. Demonstrating InitContainers

**Note:** If you don’t have an existing Kubernetes cluster, refer to the Demo 1.1 of Lesson 1.

**Step 1:** Demonstrating InitContainers

* Start the kubernetes cluster in the lab
* Let’s now create a simple pod that has two init containers. The first waits for myservice and the second waits for mydb. Once both init containers are complete, the pod runs the app container from its spec section. Use **cat > <filename>** to create a file as shown below:

*cat > myapp.yaml*

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: myservice*

*spec:*

*ports:*

*- protocol: TCP*

*port: 80*

*targetPort: 9376*

*---*

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: mydb*

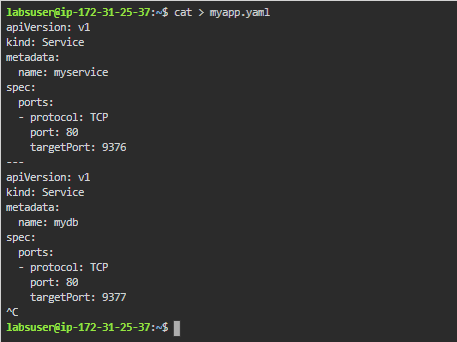
*spec:*

*ports:*

*- protocol: TCP*

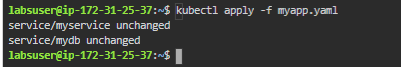
*port: 80*

*targetPort: 9377*



In the configuration file, you can see that the pod has a volume named **shared-data**

* You can now start this pod by running the kubectl command as shown below:

*kubectl apply -f myapp.yaml*

* Check on the pod’s status using ***kubectl get -f myapp.yaml*** or for getting more details, use **kubectl describe -f myapp.yaml**

