# **Lab x: Working with ConfigMap**

## **Task 1: Setting Container Environment Variables using ConfigMap**

1. Create a ConfigMap yaml and enter the contents given below

$ vi config-map.yaml



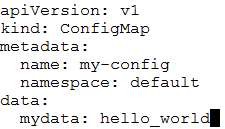
apiVersion: v1 kind: ConfigMap metadata:

name: my-config namespace: default

data:

mydata: hello\_world

Once entered, it should look as shown below



1. Create the ConfigMap using the yaml

$ kubectl create -f config-map.yaml



1. Create a pod that uses the ConfigMap using the content given below

$ vi cm-pod.yaml



apiVersion: v1 kind: Pod metadata:

name: cm-pod spec:

containers:

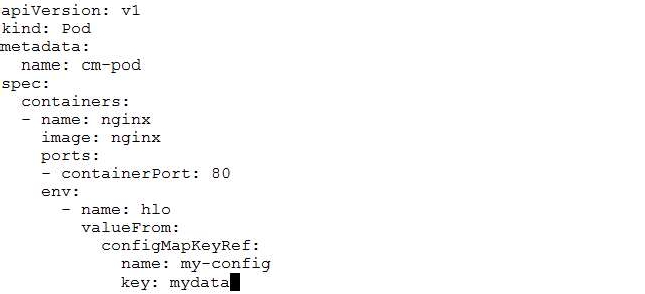
- name: nginx image: nginx ports:

- containerPort: 80 env:

- name: hlo valueFrom:

configMapKeyRef: name: my-config key: mydata

Once entered, it should look as shown below



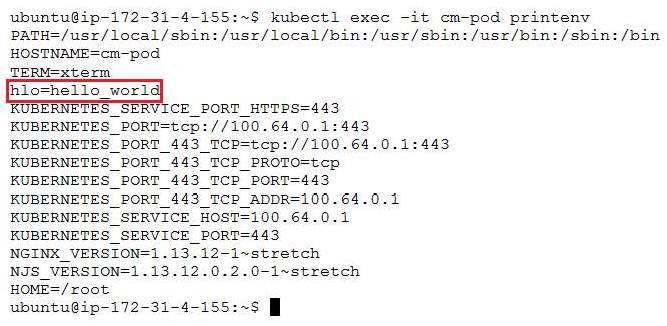
1. Create the Pod using the yaml

$ kubectl create -f cm-pod.yaml



1. Once the Pod is up, see that the environment variable specified in the ConfigMap is set in the container

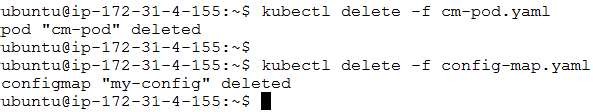
$ kubectl exec -it cm-pod printenv



1. Delete the resources created in this task

$ kubectl delete -f cm-pod.yaml

$ kubectl delete -f config-map.yaml

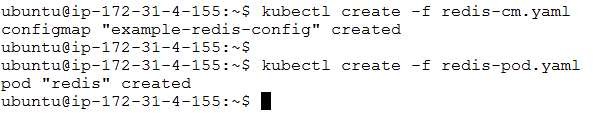


## **Task 2: Setting Configuration File with Volume using ConfigMap**

1. Download the ConfigMap and Pod yaml using the git hub links and Create the resources using the yamls downloaded in the previous step

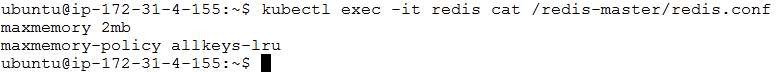
$ kubectl create -f redis-cm.yaml

$ kubectl create -f redis-pod.yaml



1. See that the config file **redis.conf** is present at **/redis-master/** and is having the contents specified in the ConfigMap

$ kubectl exec -it redis cat /redis-master/redis.conf



1. Delete all the resources created in this task

$ kubectl delete -f redis-pod.yaml

$ kubectl delete -f redis-cm.yaml

