Lesson 2 Demo 7: Create Custom Scheduler in Kubeadm

This section will guide you to:

* Create custom scheduler in Kubeadm

This lab has one sub-section, namely:

1. Creating custom scheduler in Kubeadm

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

**Step 1:** Creating custom scheduler in Kubeadm

**Note:** A Deployment manages a Replica Set which in turn manages the pods, thereby making the scheduler resilient to failures.

* Create a deployment config file. Save it as *my-scheduler.yaml*

***cat > my-scheduler.yaml***

*apiVersion: v1*

*kind: ServiceAccount*

*metadata:*

*name: my-scheduler*

*namespace: kube-system*

*---*

*apiVersion: rbac.authorization.k8s.io/v1*

*kind: ClusterRoleBinding*

*metadata:*

*name: my-scheduler-as-kube-scheduler*

*subjects:*

*- kind: ServiceAccount*

*name: my-scheduler*

*namespace: kube-system*

*roleRef:*

*kind: ClusterRole*

*name: system:kube-scheduler*

*apiGroup: rbac.authorization.k8s.io*

*---*

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*labels:*

*component: scheduler*

*tier: control-plane*

*name: my-scheduler*

*namespace: kube-system*

*spec:*

*selector:*

*matchLabels:*

*component: scheduler*

*tier: control-plane*

*replicas: 1*

*template:*

*metadata:*

*labels:*

*component: scheduler*

*tier: control-plane*

*version: second*

*spec:*

*serviceAccountName: my-scheduler*

*containers:*

*- command:*

*- /usr/local/bin/kube-scheduler*

*- --address=0.0.0.0*

*- --leader-elect=false*

*- --scheduler-name=my-scheduler*

*image: gcr.io/my-gcp-project/my-kube-scheduler:1.0*

*livenessProbe:*

*httpGet:*

*path: /healthz*

*port: 10251*

*initialDelaySeconds: 15*

*name: kube-second-scheduler*

*readinessProbe:*

*httpGet:*

*path: /healthz*

*port: 10251*

*resources:*

*requests:*

*cpu: '0.1'*

*securityContext:*

*privileged: false*

*volumeMounts: []*

*hostNetwork: false*

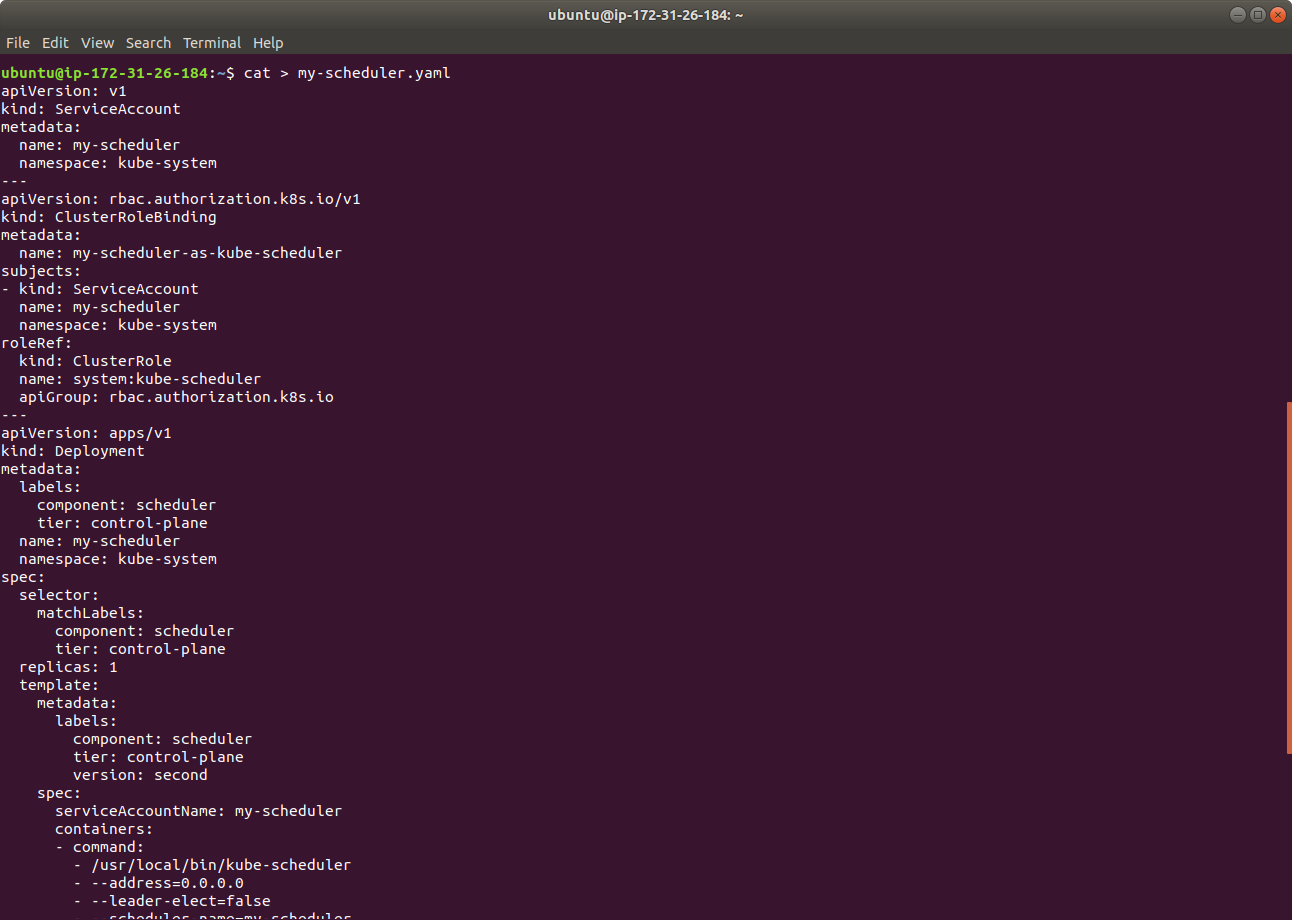
*hostPID: false*

*volumes: []*

* In order to run your scheduler in a Kubernetes cluster, just create the deployment specified in the config above in a Kubernetes cluster:

*kubectl create -f my-scheduler.yaml*

(Please Note: You might already have this file applied from the previous demos)



* Verify that the scheduler pod is running:

*kubectl get pods --namespace=kube-system*

