**5. Create Kubernetes Liveness & Readiness Probes**

--- **note** – we are going to see, what are liveness and rediness probe and how we are going to define them in our user management application.

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/05-Kubernetes-Important-Concepts-for-Application-Deployments/05-03-Kubernetes-Liveness-and-Readiness-Probes>

--- **References**: <https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/>

**Introduction**

**Create Liveness Probe with Command**

           livenessProbe:

            exec:

              command:

                - /bin/sh

                - -c

                - nc -z localhost 8095

            initialDelaySeconds: 60

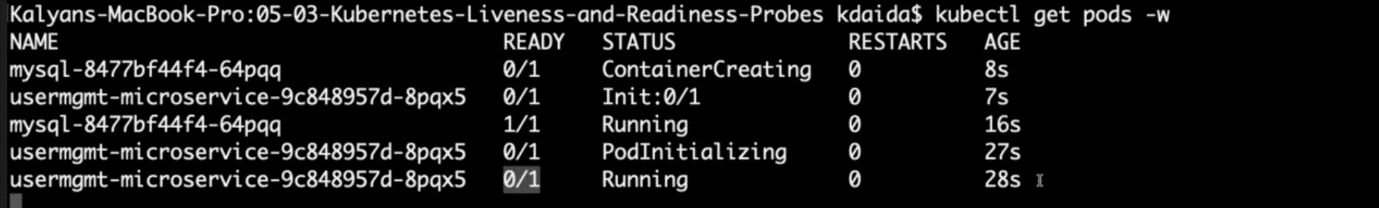
            periodSeconds: 10

--- **initialDelaySeconds** – the amount of time after the container started and then rediness and liveness to be triggered or how much time it needs to be waited before triggering these probes. The reason for this once we start the application, just the application to be settle down, we can give some amount of delay. Default is 0 seconds, here I am giving 60 seconds.

--- until these 60 seconds going to be completed, you’re not going to see that the pod is running. The pod will not come to the ready state. Now I put 60 seconds but based on our application requirement you can put 10 or 20 0r 30.

--- **note** - because of the rediness probe, our application user management micro service will start after 60 seconds.

--- **kubectl get pods -w**



--- **note** - because of the rediness probe, our user management microservice application is not coming even after 28 seconds, our application comes to live only after 60 seconds. Based on our application we can mention the **initialDelaySeconds= 10 or 20 or 30** …etc.

**Create Readiness Probe with HTTP GET**

          readinessProbe:

            httpGet:

              path: /usermgmt/health-status # the url we used to see the application on the browser.

              port: 8095

            initialDelaySeconds: 60

            periodSeconds: 10

--- **06-UserManagementMicroservice-Deployment-Service.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: usermgmt-microservice

  labels:

    app: usermgmt-restapp

spec:

  replicas: 1

  selector:

    matchLabels:

      app: usermgmt-restapp

  template:

    metadata:

      labels:

        app: usermgmt-restapp

    spec:

      initContainers:

        - name: init-db

          image: busybox:1.31

          command: ['sh', '-c', 'echo -e "Checking for the availability of MySQL Server deployment"; while ! nc -z mysql 3306; do sleep 1; printf "-"; done; echo -e "  >> MySQL DB Server has started";']

      containers:

        - name: usermgmt-restapp

          image: stacksimplify/kube-usermanagement-microservice:1.0.0

          ports:

            - containerPort: 8095

          env:

            - name: DB\_HOSTNAME

              value: "mysql"

            - name: DB\_PORT

              value: "3306"

            - name: DB\_NAME

              value: "usermgmt"

            - name: DB\_USERNAME

              value: "root"

            - name: DB\_PASSWORD

              valueFrom:

                secretKeyRef:

                  name: mysql-db-password

                  key: db-password

          livenessProbe:

            exec:

              command:

                - /bin/sh

                - -c

                - nc -z localhost 8095

            initialDelaySeconds: 60

            periodSeconds: 10

          readinessProbe:

            httpGet:

              path: /usermgmt/health-status

              port: 8095

            initialDelaySeconds: 60

            periodSeconds: 1

**Create k8s objects & Test**

**# Create All Objects**

--- **kubectl apply -f kube-manifests/**

**# List Pods**

--- **kubectl get pods**

**# Watch List Pods screen**

--- **kubectl get pods -w**

**# Describe Pod & Discuss about init container**

--- **kubectl describe pod <usermgmt-microservice-xxxxxx>**

**# Access Application Health Status Page**

--- **http://<WorkerNode-Public-IP>:31231/usermgmt/health-status**

--- **Observation**: User Management Microservice pod witll not be in READY state to accept traffic until it completes the **initialDelaySeconds=60seconds**.

**Clean-Up**

--- Delete all k8s objects created as part of this section

**# Delete All**

--- **kubectl delete -f kube-manifests/**

**# List Pods**

--- **kubectl get pods**

**# Verify sc, pvc, pv**

--- **kubectl get sc,pvc,pv**