K8s deployment:

Imp Links:

Liveness,readiness,startup probes

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

<https://www.youtube.com/watch?v=aTlQBofihJQ>

Images:

<https://kubernetes.io/docs/concepts/containers/images/#pre-pulled-images>

Gracefulshutdown:

<https://pracucci.com/graceful-shutdown-of-kubernetes-pods.html#:~:text=yaml%20shows%20a%20simple%20deployment,60%20seconds%20termination%20grace%20period>.

Serviceaccunt:

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/>

restartpolicy:

<https://gaurav-kaushikgk88.medium.com/self-healing-pods-in-kubernetes-their-restart-policies-84d929778e6d>

rolling updates:

<https://www.bluematador.com/blog/kubernetes-deployments-rolling-update-configuration>

<https://www.weave.works/blog/kubernetes-deployment-strategies>

node affinity:

<https://www.youtube.com/watch?v=I7t57_tQBz4>

myDeployment manifest:

apiVersion: apps/v1

kind: Deployment

metadata:

  labels:

    environment: calc-env #used to manage replica sets and pods with matching env label

  name: calc-app

spec:

  affinity:

    nodeAffinity: # places our pods in a particular node based on march expression

      requiredDuringSchedulingIgnoredDuringExecution:

        nodeSelectorTerms:

        - matchExpressions:

          - key: gpu #a node should be there with gpu label

            operator: Exists #gpu key should exist dont care about the gpu value

  replicas: 3 #making 3 replicas

  selector:

    matchLabels:

      app: calc-app

      enveronment: calc-env

  minReadySeconds: 10

  strategy:

    type: RollingUpdate #for updating our replica set(recreate will terminates all pods and before adding new)

    rollingUpdate:

      maxUnavailable: 1 #max only one pod should be unavailabele and 2 should be avail

      maxSurge: 1 #at a time one pod should be updated

  template:

    metadata:

      labels:

        app: calc-app

        environment: calc-env

    spec:

      terminationGracePeriodSeconds: 10 #shutdowns containers usecase1 while rolling updates

      serviceAccountName: default #Processes in containers in pods can contact the apiserver they are authenticated as a particular Service Account

      restartPolicy: OnFailure #restarts pod if its unhealthy

      containers:

      - name: calc-app

        image: venkatrobin/flask-demo:v1

        imagePullPolicy: Never

        resources:

          limits:

            memory: "128Mi"

            cpu: "500m"

        ports:

        - containerPort: 8080

        readinessProbe: # is pod ready to receive traffic probe will tell weather pod is heatlthy or no to cluster

          httpGet:

          path: /health #this is endpoint url path to our health status

          port: 8080

          initialDelaySeconds: 15 #checks first time after creation of container after 15 sec

          periodSeconds: 10 #checks every 10 seconds regularly

        livenessProbe: #checks for deadlock if present restarts the container

          httpGet:

          path: /health #this is endpoint url path to our health status

          port: 8080

          initialDelaySeconds: 15 #checks first time after creation of container after 15 sec

          periodSeconds: 10 # checks every 10 seconds regularly

        startupProbe: #checks for sucessfull startup if failed readiness and liveness wont trigger

          httpGet:

          path: /health

          port: 8080

          failureThreshold: 30 #max amount of time for the app to finish the startup (30\*10=5mins)

          periodSeconds: 10

---

apiVersion: v1

kind: Service

metadata:

  name: calc-app

spec:

  selector:

    app: calc-app

  ports:

  - port: 8080

    targetPort: 5000