**2. Kubernetes Secrets**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/05-Kubernetes-Important-Concepts-for-Application-Deployments/05-01-Kubernetes-Secrets>

**Introduction**

--- Kubernetes Secrets let you store and manage sensitive information, such as passwords, OAuth tokens, and ssh keys.

--- Storing confidential information in a Secret is safer and more flexible than putting it directly in a Pod definition or in a container image.

**Create Secret for MySQL DB Password**

**# Mac**

--- **echo -n 'dbpassword11' | base64**

**# URL: https://www.base64encode.org**

--- **note** - Create Kubernetes Secrets manifest

apiVersion: v1

kind: Secret

metadata:

  name: mysql-db-password

#type: Opaque means that from Kubernetes’s point of view the contents of this Secret is unstructured.

#It can contain arbitrary key-value pairs.

type: Opaque

data:

  # Output of echo -n 'dbpassword11' | base64

  db-password: ZGJwYXNzd29yZDEx

**Update secret in MySQL Deployment for DB Password**

          env:

            - name: MYSQL\_ROOT\_PASSWORD

              valueFrom:

                secretKeyRef:

                  name: mysql-db-password

                  key: db-password

**Update secret in UMS Deployment**

--- UMS means User Management Microservice

            - name: DB\_PASSWORD

              valueFrom:

                secretKeyRef:

                  name: mysql-db-password

                  key: db-password

**Create & Test**

**# Create All Objects**

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

**# List Pods**

--- **kubectl get pods**

**# Access Application Health Status Page**

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

**Clean-Up**

--- **note** - 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** -