ASSIGNMENT 3

Exercise 5: Create and Use a Secret

Objective: Store sensitive data using Secrets and use it in a pod

Step 1: Create a Secret for database credentials:

```
master@master-vm:~$ kubectl create secret generic db-secret --from-literal=DB_USER=admin
secret/db-secret created
```

Step 2: Verify the Secret:

```
master@master-vm:~$ kubectl create secret generic db-secret \
>    --from-literal=DB_USER=myusername \
>    --from-literal=DB_PASS=mypassword \
>    --dry-run=client -o yaml | kubectl apply -f -
Warning: resource secrets/db-secret is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by
kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubect
l apply. The missing annotation will be patched automatically.
secret/db-secret configured
```

```
master@master-vm:~$ kubectl describe secret db-secret
Name:         db-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
DB_PASS:  10 bytes
DB_USER:  10 bytes
```

Step 3: Create a Pod that uses the Secret (`nginx-secret-pod.yaml`):

```
master@master-vm:~$ nano nginx-secret-pod.yaml
```

```
  GNU nano 4.8                                                   nginx-s
apiVersion: v1
kind: Pod
metadata:
  name: nginx-secret-pod
spec:
  containers:
    - name: nginx
      image: nginx
      env:
        - name: DB_USER
          valueFrom:
            secretKeyRef:
              name: db-secret
              key: DB_USER
        - name: DB_PASS
          valueFrom:
            secretKeyRef:
              name: db-secret
              key: DB_PASS
```

Step 4: Deploy the pod:

```
master@master-vm:~$ kubectl apply -f nginx-secret-pod.yaml
pod/nginx-secret-pod unchanged
```

Step 5: Check the pod and logs:

```
master@master-vm:~$ kubectl exec -it nginx-secret-pod -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=nginx-secret-pod
TERM=xterm
DB_USER=myusername
DB_PASS=mypassword
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
NGINX_VERSION=1.27.4
NJS_VERSION=0.8.9
NJS_RELEASE=1~bookworm
PKG_RELEASE=1~bookworm
DYNPKG_RELEASE=1~bookworm
HOME=/root
```

Step 6: Delete the pod and Secret:

Exercise 6: Create and Expose a Service

Objective: Deploy an application and expose it using a service.

Step 1: Create a deployment:

Step 2: Expose the deployment using a service:

```
master@master-vm:~$ kubectl expose deployment webapp --type=NodePort --port=80
service/webapp exposed
```

Step 3: **Get service details:**

```
master@master-vm:~$ kubectl get svc webapp
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
webapp    NodePort   10.102.63.54   <none>        80:30771/TCP   19s
```

Step 4: Access the service (Minikube users):

```
master@master-vm:~$ minikube service webapp --url
http://192.168.49.2:30771
```

```
master@master-vm:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
webapp-6fddc68b96-5nwwt   1/1     Running   0          2m53s
```

Step 5: Delete the service and deployment:

```
master@master-vm:~$ kubectl delete svc webapp
service "webapp" deleted
master@master-vm:~$ kubectl delete deployment webapp
deployment.apps "webapp" deleted
```