Lab Exercise 12- Creating Secrets in Kubernetes

Creating secrets in Kubernetes is an essential practice for managing sensitive information such as passwords, OAuth tokens, and SSH keys. Let's go through a lab exercise that demonstrates how to create different types of secrets in Kubernetes.

Step 1: Create a Generic Secret

Create a secret from literal values:

```
kubectl create secret generic my-secret --from-literal=username=admin --from-literal=password='P@ssword'
```

Verify the secret:

```
kubectl get secrets my-secret -o yaml
```

The output will be base64 encoded. To decode, you can use:

```
echo -n “P@ssword” | base64
```

Step 2: Create a Secret from a File

Create a file with your sensitive data:

```
echo -n 'my-username' > ./username.txt
echo -n 'my-password' > ./password.txt
```

Create a secret from the file:

```
kubectl create secret generic my-file-secret --from-file=username=./username.txt --from-file=password=./password.txt
```

Verify the secret:

```
kubectl get secrets my-file-secret -o yaml
```

Step 3: Create a Docker Registry Secret

Create a Docker registry secret:

```
kubectl create secret docker-registry my-registry-secret --docker-username=<your-username> --docker-password=<your-password> --docker-email=<your-email> --docker-server=<your-docker-server>
```

Verify the secret:

```
kubectl get secrets my-registry-secret -o yaml
```

Step 4: Create a Secret Using a YAML Manifest

Create a YAML file for the secret:

Create a file named **my-secret-pod.yaml** with the following content:

```
apiVersion: v1
kind: Secret
metadata:
  name: my-manifest-secret
type: Opaque
```

```
data:
  username: YWRtaW4=    # base64 encoded value of 'admin'
  password: UEBzc3cwcmQ=  # base64 encoded value of 'P@ssword'
```

Apply the YAML file:

```
kubectl apply -f my-manifest-secret.yaml
```

Verify the secret:

```
kubectl get secrets my-manifest-secret -o yaml
```

Step 5: Use Secrets in Pods

Create a pod that uses the secret:

Create a file named **pod-with-secret.yaml** with the following content:

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-secret
spec:
  containers:
  - name: mycontainer
    image: nginx
    env:
    - name: USERNAME
      valueFrom:
        secretKeyRef:
          name: my-secret
```

```
      key: username
  - name: PASSWORD
    valueFrom:
      secretKeyRef:
        name: my-secret
        key: password
 restartPolicy: Never
```

Apply the YAML file:

```
kubectl apply -f pod-with-secret.yaml
```

Verify the pod is running and the secret is being used:

```
kubectl exec pod-with-secret -- printenv | grep -E 'USERNAME|PASSWORD'
```

Step 7: Clean Up

Delete the secrets:

```
kubectl delete secret my-secret
kubectl delete secret my-file-secret
kubectl delete secret my-registry-secret
kubectl delete secret my-manifest-secret
```

Delete the pod:

```
kubectl delete pod pod-with-secret
```